

# **LAPORAN AKHIR**

## **PROYEK**



ADMINISTRASI SISTEM APLIKASI DAN DATABASE PADA SQL SERVER 2008

Disusun Oleh:

Nama : 11117320

NPM : Chantika Amanda

Kelas :

**LEMBAGA PENGEMBANGAN KOMPUTERISASI**  
**UNIVERSITAS GUNADARMA**

**2022**

# DAFTAR ISI

DESKRIPSI LAPORAN .....	3
TAHAPAN Pengerjaan.....	4
PENUTUP .....	7



# DESKRIPSI LAPORAN

ADMINISTRASI SISTEM APLIKASI DAN DATABASE PADA SQL SERVER  
2008

laporan berisikan cara melakukan instalasi SQL Server 2008, mengetahui tahap instalasi, bekerja dalam lingkungan SQL Server Management Studio dan Query Editor, Objek pada SQL Server 2008, dan cara pembuatan database menggunakan SQL Server 2008 dan tahap pengerjaan dalam menyelesaikan tugas dalam Materi.



# TAHAPAN Pengerjaan

## 1. Ringkasan Materi

Microsoft SQL Server 2008 dan Kebutuhan Instalasinya SQL Server pada awalnya adalah versi pembaharuan dari database relasi Sybase. Ini hanya tersedia pada platform Windows. Versi sekarang (Juli 2010) adalah SQL Server 2008 R2. Terdapat dua perbedaan utama antara SQL Server / Sybase dan Oracle adalah:

- SQL Server tidak memiliki segmen undo atau rollback dan oleh karena itu tidak ada fungsi rollback / commit, atau membaca tampilan data yang konsisten, selain yang sedikit bisa ditampung dari on line redo logs.
- Arsitektur SQL Server adalah: 1 mesin host ke 1 atau lebih Contoh untuk banyak Database.

SQL Server memiliki komponen opsional yang tidak dikenakan biaya tambahan (jika dihosting pada mesin yang sama), yaitu:

- Pelaporan SQL Server Services Business Intelligence reporting, alternatif Microsoft untuk OBIEE, Discoverer, Crystal Reports. Memiliki integrasi yang sangat baik dengan produk klien Microsoft seperti Excel yang signifikan bagi kegunaan pengguna akhir.
- Analysis Services OLAP (on line analytical processing), cubes, data mining.
- SQL Server Integration Services ETL (extract transform load) dan ekspor / impor tool. Sama dengan ekspor / impor, data pump, sql loader, skrip shell DBA. Ini disebut DTS (Data Transformation Services) di versi lama.

Mode Otentikasi yang Dipilih Sehubungan dengan Database Engine, ada dua mode otentikasi yang berbeda:

- Windows mode Spesifikasi security secara eksklusif pada tingkat sistem operasi yaitu, ia akan menentukan bagaimana pengguna terhubung ke sistem operasi Windows menggunakan akun pengguna dan keanggotaan grup mereka.
- Mixed mode Memungkinkan pengguna untuk terhubung ke Database Engine menggunakan otentikasi Windows atau otentikasi SQL Server. Ini berarti beberapa akun pengguna dapat disiapkan untuk menggunakan subsistem keamanan Windows, sementara yang lain dapat menggunakan subsistem keamanan SQL Server di samping subsistem keamanan Windows.

Objek-objek SQL Server 2008 :

- Database  
Database merupakan sekumpulan data yang terorganisir. Terdiri dari satu atau lebih tabel.
- Tabel  
Tabel merupakan kumpulan dari baris/record dan kolom/field.
- Database Diagram  
Diagram database adalah representasi grafik dari sebuah database.
- View  
View adalah table virtual yang isinya didefinisikan oleh query database. View bukanlah sebuah table fisik, tetapi sekumpulan instruksi yang menghasilkan sekumpulan data.
- Indeks  
Indeks adalah sebuah file pelengkap yang berhubungan dengan table, yang fungsinya mempercepat waktu akses ke dalam baris-baris table. Indeks dibentuk oleh kunci-kunci berdasarkan kepada isi dari satu table atau lebih kolom pada table.
- Stored Procedure  
Sebuah program yang ditulis dalam bahasa T-SQL yang di simpan di dalam database SQL Server. Bisa dijalankan secara manual atau dijalankan oleh program lain.

Dalam pembentukan Database ada 3 hal yang perlu kita perhatikan :

- Penamaan Database itu sendiri. Disini meliputi nama database itu serta aturan penggunaan huruf (collation) yang digunakan.
- Data File. File tempat menampung data dari database.
- Transaction Log. Tempat untuk mencatat setiap kegiatan yang dilakukan terhadap database dan penampungan perubahan data.

Saat Database di buat, ada tiga file yang dapat dibuat oleh SQL Server, file-file tersebut yaitu :

- File Primer (.MDF) Mengandung informasi yang dibutuhkan untuk menu shortcut-load dan menginisialisasi database serta untuk meyimpan data.
- File Sekunder (.NDF) File ini ada jika file primer tidak cukup bisa memelihara semua file data.
- File Log (.LDF) File ini berfungsi untuk memulihkan database. Ukuran minimalnya 512 KB.

## **2. Langkah-Langkah**

1. Download SQL Server Installer melalui Website Microsoft
2. Install SQL Server

SQL Server 2008 R2 Setup

## Instance Configuration

Specify the name and instance ID for the instance of SQL Server. Instance ID becomes part of the installation path.

Setup Support Rules  
License Terms  
Feature Selection  
Installation Rules  
**Instance Configuration**  
Disk Space Requirements  
Server Configuration  
Database Engine Configuration  
Error Reporting  
Installation Configuration Rules  
Installation Progress  
Complete

☐ Default instance

☒ Named instance:

---

Instance ID:

Instance root directory:  ...

---

SQL Server directory: C:\Program Files\Microsoft SQL Server\MSSQL10\_50.SQLEXPRESS

Installed instances:

Instance Name	Instance ID	Features	Edition	Version
<Shared Compon...		SSMS		10.52.4000.0

< Back
Next >
Cancel
Help

### 3. Pilih Mixed Mode untuk Otentikasi dan masukan Password

SQL Server 2008 R2 Setup

## Database Engine Configuration

Specify Database Engine authentication security mode, administrators and data directories.

Setup Support Rules  
License Terms  
Feature Selection  
Installation Rules  
Instance Configuration  
Disk Space Requirements  
Server Configuration  
**Database Engine Configuration**  
Error Reporting  
Installation Configuration Rules  
Installation Progress  
Complete

Account Provisioning | Data Directories | User Instances | FILESTREAM

Specify the authentication mode and administrators for the Database Engine.

Authentication Mode

☐ Windows authentication mode

☒ Mixed Mode (SQL Server authentication and Windows authentication)

Specify the password for the SQL Server system administrator (sa) account.

Enter password:

Confirm password:

Specify SQL Server administrators

### 4. Buat Database

```
CREATE DATABASE lepkom_mandiri_11117320
```

# **PENUTUP**

## **1. Kesimpulan**

Pada bab ini telah dipelajari cara melakukan instalasi SQL Server dan juga Mempelajari mengenai SSMS, Objek database cara membuat dan modifikasi Database serta Metode Otentikasi untuk masuk kedalam Server Database

## **2. SaranPenulisan**



# **LAPORAN AKHIR**

## **PROYEK**



DML (DATA MANIPULATION LANGUAGE) DAN JOIN TABLE

Disusun Oleh:

Nama : 11117320

NPM : Chantika Amanda

Kelas :

**LEMBAGA PENGEMBANGAN KOMPUTERISASI**  
**UNIVERSITAS GUNADARMA**

**2022**



# DAFTAR ISI

DESKRIPSI LAPORAN .....	3
TAHAPAN Pengerjaan.....	4
PENUTUP .....	17



# **DESKRIPSI LAPORAN**

## **DML (DATA MANIPULATION LANGUAGE) DAN JOIN TABLE**

Pada pertemuan 2 laporan berisikan mengenai cara melakukan DML seperti SELECT, INSERT, UPDATE, DELETE selain itu seperti cara membuat Table, Tipe Data, Database Diagram, Constraint yang ada pada Table, Join Table



# TAHAPAN Pengerjaan

## 1. Ringkasan Materi

DML (Data Manipulation Language) digunakan untuk memanipulasi data yang ada dalam suatu table:

- SELECT untuk menampilkan data
- INSERT untuk menambahkan data baru
- UPDATE untuk mengubah data yang sudah ada
- DELETE untuk menghapus data

SQL Server menyediakan banyak cara untuk bekerja dengan tabel. Membuat tabel baru dengan menggunakan fitur New Table di SQL Server Management Studio menggunakan perintah CREATE TABLE, memodifikasi tabel yang ada menggunakan fitur Modify Table di SQL Server Management Studio dengan perintah ALTER TABLE, serta melakukan fungsi manajemen table lainnya, termasuk menyalin, mengganti nama, dan menghapus tabel.

Sebelum membuat tabel, perlu diperhatikan dalam penggunaan nama tabel. Nama tabel dapat memuat sampai 128 karakter. Nama tabel harus dimulai dengan karakter alfabet dan mengandung underscore (\_), simbol (@), tanda pagar (#) dan nomor. Pengecualian di dalam aturan ini terdapat pada tabel temporary. Tabel temporary local, memiliki nama yang dimulai dengan # dan di akses hanya ke beberapa sesi pengguna. Tabel temporary global memiliki nama yang dimulai dengan ## dan di akses kepada siapa pun selama sesi pengguna tetap terhubung. Tabel temporary dibuat di dalam database tempdb dan dihapus secara otomatis saat sesi pengguna berakhir. Nama tabel harus unik untuk setiap skema di dalam sebuah database. Namun skema dapat berisi like-named tables. Sehingga dapat membuat tabel multiple contacts selama file tersebut didefinisikan dalam skema terpisah. Contohnya, Customer, Employee, dan Servis yang dimiliki oleh tabel contacts

Beberapa Constraint yang sering digunakan

- NOT NULL Menentukan bahwa kolom tidak dapat menerima nilai null. Namun, nilai null bukan nilai 0, tetapi nilai null adalah yang tidak diisi atau tidak terdefinisi.
- Primary Key Satu atau lebih kolom pada tabel yang memiliki baris yang eksklusif di dalam tabel akan membentuk primary key. Spesifikasi dari primary key memastikan integritas dari tabel. Kolom yang membentuk primary key tidak bisa mengandung nilai null. Apabila menentukan nilai

sebagai primary, SQL Server akan membuat indeks tertentu untuk memastikan bahwa kunci tersebut bersifat eksklusif. Tabel yang merupakan primary key ditandai dengan adanya tanda kunci disamping kolom.

- Foreign Key Menentukan hubungan antara tabel-tabel. Foreign Key adalah field di table aktif yang menunjukan ke field kunci pada tabel lain. Penggunaan foreign key mampu mencegah penghapusan baris-baris pada tabel yang aktif, apabila ada referensi field-field kunci dari tabel eksternal. Batasan semacam ini mencegah nilai yang dimasukkan ke dalam tabel yang tidak ditentukan pada field kunci dari tabel eksternal yang digabungkan.

Diagram database adalah representasi dari sebuah database. Dengan diagram database bisa mengelola database dengan menggunakan antarmuka grafis serta menggeser mouse untuk melakukan tugas-tugas yang biasanya dilakukan oleh perintah bahasa SQL. Tugas-tugas ini meliputi menambah atau menghapus tabel, trigger prosedur, batasan kolom tabel, serta hubungan antar tabel.

JOIN merupakan salah satu operasi untuk menghubungkan dua atau lebih tabel yang saling berelasi. JOIN sendiri memiliki berbagai jenis, diantaranya : JOIN, INNER JOIN, LEFT JOIN, LEFT OUTER JOIN, RIGHT JOIN, RIGHT OUTER JOIN, CROSS JOIN, dan JOIN dengan koma.

- JOIN / INNER JOIN merupakan jenis JOIN yang digunakan untuk mendapatkan data dari dua tabel atau lebih yang persis saling berelasi. Hasil dari JOIN dengan menggunakan JOIN / INNER JOIN merupakan irisan dari relasi antar tabel.
- LEFT JOIN / LEFT OUTER JOIN merupakan jenis JOIN yang digunakan untuk mendapatkan data dari dua tabel atau lebih dimana data di tabel sebelah kiri ditampilkan semua baik yang berelasi dengan data di tabel sebelah kanan maupun tidak
- RIGHT JOIN / RIGHT OUTER JOIN merupakan jenis JOIN yang digunakan untuk mendapatkan data dari dua tabel atau lebih dimana data di tabel sebelah kanan ditampilkan semua baik yang berelasi dengan data di tabel sebelah kiri maupun tidak
- CROSS JOIN merupakan jenis JOIN yang digunakan untuk mendapatkan data kombinasi dari dua tabel atau lebih. Misalkan,  $n$  = jumlah baris data pada tabel di sebelah kiri, dan  $m$  = jumlah baris data pada tabel di sebelah kanan. Maka hasil jumlah baris dari CROSS JOIN adalah  $n \times m$  baris data.
- Selain menggunakan operator JOIN, dapat juga melakukan JOIN antar tabel dengan menggunakan koma. Bila tidak menambahkan kondisi relasi dengan menggunakan WHERE, maka JOIN dengan koma akan menghasilkan data yang sama apabila menggunakan CROSS JOIN

## **2. Langkah-Langkah**

1. Buat Table (Script Terlampir pada Zip File)

```

5
6
7 -- No 1
8
9 CREATE TABLE regions
10 ( region_id      NUMERIC(3) PRIMARY KEY
11   CONSTRAINT region_id_nn NOT NULL
12   , region_name   VARCHAR(25)
13 );
14
15 CREATE TABLE countries
16 ( country_id     CHAR(2)
17   CONSTRAINT country_id_nn NOT NULL
18   , country_name  VARCHAR(40)
19   , region_id     NUMERIC
20   CONSTRAINT country_c_id_pk
21                 PRIMARY KEY (country_id)
22 );
23
24 CREATE TABLE locations
25 ( location_id    NUMERIC(4)
26   CONSTRAINT location_id_nn NOT NULL
27   , street_address VARCHAR(40)
28   , postal_code   VARCHAR(12)
29   , city          VARCHAR(30)

```

## 2. Masukkan Data

```

87
88 -- No 2
89
90 INSERT INTO regions VALUES
91 ( 1
92   , 'Europe'
93 );
94
95 INSERT INTO regions VALUES
96 ( 2
97   , 'Americas'
98 );
99
100 INSERT INTO regions VALUES
101 ( 3
102   , 'Asia'
103 );
104
105 INSERT INTO regions VALUES
106 ( 4
107   , 'Middle East and Africa'
108 );
109 select * from regions;
110
111 INSERT INTO countries VALUES
112 ( 'IT'
113   , 'Italy'

```

3. Tampilkan data karyawan yang memiliki nama berawalan huruf "M". Dengan menggunakan Clause WHERE LIKE maka kita dapat menampilkan Karyawan yang berawalan dengan M

```
2389  
2390 SELECT * FROM employees WHERE first_name LIKE 'M%'  
2391  
2392
```

	employee_id	first_name	last_name	email	phone_NUMERIC	hire_date	job_id	salary	commission_pct	manager_id	department_id
1	120	Matthew	Weiss	MWEISS	650.123.1234	1996-07-18	ST_MAN	8000.00	NULL	100	50
2	130	Mozhe	Atkinson	MATKINSO	650.124.6234	1997-10-30	ST_CLERK	2800.00	NULL	121	50
3	134	Michael	Rogers	MROGERS	650.127.1834	1998-08-26	ST_CLERK	2900.00	NULL	122	50
4	164	Mattea	Marvins	MMARVINS	011.44.1346.329268	2000-01-24	SA_REP	7200.00	0.10	147	80
5	182	Martha	Sullivan	MSULLIVA	650.507.9878	1999-06-21	SH_CLERK	2500.00	NULL	120	50
6	201	Michael	Hartstein	MHARTSTE	515.123.5555	1996-02-17	MK_MAN	13000.00	NULL	100	20

4. Tampilkan data karyawan yang mulai bekerja antara tanggal 20 Mei 1991 dan 17 Juni 1995. Dengan menggunakan Clause WHERE dan BETWEEN kita dapat mencari suatu nilai diantara 2 nilai lain

```
2392  
2393 SELECT * FROM employees WHERE hire_date BETWEEN '1991/05/20' AND '1995/06/17'
```

	employee_id	first_name	last_name	email	phone_NUMERIC	hire_date	job_id	salary	commission_pct	manager_id	department_id
1	102	Lex	De Haan	LDEHAAN	515.123.4569	1993-01-13	AD_VP	17000.00	NULL	100	90
2	104	Bruce	Ernst	BERNST	590.423.4568	1991-05-21	IT_PROG	6000.00	NULL	103	60
3	108	Nancy	Greenberg	NGREENBE	515.124.4569	1994-08-17	FI_MGR	12000.00	NULL	101	100
4	109	Daniel	Faviet	DFAVIET	515.124.4169	1994-08-16	FI_ACCOUNT	9000.00	NULL	108	100
5	114	Den	Raphaely	DRAPHEAL	515.127.4561	1994-12-07	PU_MAN	11000.00	NULL	100	30
6	115	Alexander	Khoo	AKHOO	515.127.4562	1995-05-18	PU_CLERK	3100.00	NULL	114	30
7	122	Payam	Kaufling	PKAUFLIN	650.123.3234	1995-05-01	ST_MAN	7900.00	NULL	100	50
8	203	Susan	Mavris	SMAVRIS	515.123.7777	1994-06-07	HR_REP	6500.00	NULL	101	40
9	204	Hermann	Baer	HBAER	515.123.8888	1994-06-07	PR_REP	10000.00	NULL	101	70
10	205	Shelley	Higgins	SHIGGINS	515.123.8080	1994-06-07	AC_MGR	12000.00	NULL	101	110
11	206	William	Gietz	WGIEZT	515.123.8181	1994-06-07	AC_ACCOU...	8300.00	NULL	205	110

5. Ganti tipe data region\_id menjadi numeric(3) pada tabel countries, kemudian jadikan sebagai foreign key. Alter Pertama untuk mengubah tipe data menjadi Numeric(3) lalu jalankan Alter kedua yang menambahkan Constraint Foreign Key

```
2402
2403 -- No 5
2404
2405 ALTER TABLE countries
2406 ALTER COLUMN region_id NUMERIC(3)
2407 GO
2408 ALTER TABLE countries
2409 ADD CONSTRAINT countr_reg_fk FOREIGN KEY (region_id) REFERENCES regions(region_id)
2410
2411
2412 -- No 6
```

Messages  
Command(s) completed successfully.

6. Gunakan inner join untuk menampilkan region\_name dan country\_name pada tabel regions dan countries. Inner Join dengan Syarat region\_id untuk menggabungkan data 2 table

```
2416
2417 -- No 6
2418
2419 SELECT region_name, country_name FROM regions A JOIN countries B ON A.region_id = b.region_id
```

Results

	region_name	country_name
1	Americas	Argentina
2	Asia	Australia
3	Europe	Belgium
4	Americas	Brazil
5	Americas	Canada
6	Europe	Switzerland
7	Asia	China
8	Europe	Germany
9	Europe	Denmark
10	Middle East and Africa	Egypt
11	Europe	France
12	Asia	HongKong
13	Middle East and Africa	Israel
14	Asia	India
15	Europe	Italy
16	Asia	Japan
17	Middle East and Africa	Kuwait
18	Americas	Mexico

7. Buat primary key pada location\_id di tabel locations serta foreign key di country\_id. Alter table untuk menambahkan Constraint Primary Key lalu dilanjutkan penambahan Foreign Key

```
2420  
2421 ALTER TABLE locations  
2422 ADD CONSTRAINT loc_id_pk PRIMARY KEY (location_id),  
2423 CONSTRAINT loc_c_id_fk FOREIGN KEY (country_id) REFERENCES countries(country_id);
```

Messages

Command(s) completed successfully.

8. Tampilkan city dan street\_address pada tabel locations, serta country\_name pada tabel countries dengan menggunakan left outer join urutkan secara ascending berdasarkan city pada tabel locations. Dengan Menggunakan LEFT OUTER JOIN untuk menampilkan seluruh data table kiri dan data bersinggungan dengan table kanan dan lalu dengan ORDER BY kita dapat mengurutkan urutan data yang muncul berdasarkan kolom





```

2426
2427 SELECT city, street_address, country_name
2428 FROM locations A LEFT OUTER JOIN countries B ON A.country_id = B.country_id
2429 ORDER BY A.city ASC

```

Results Messages			
	city	street_address	country_name
1	Beijing	40-5-12 Laogianggen	China
2	Bern	Murtenstrasse 921	Switzerland
3	Bombay	1298 Vileparle (E)	India
4	Geneva	20 Rue des Corps-Saints	Switzerland
5	Hiroshima	9450 Kamiya-cho	Japan
6	London	8204 Arthur St	United Kingdom
7	Mexico City	Mariano Escobedo 9991	Mexico
8	Munich	Schwanthalerstr. 7031	Germany
9	Oxford	Magdalen Centre, The Oxford Science Park	United Kingdom
10	Roma	1297 Via Cola di Rie	Italy
11	Sao Paulo	Rua Frei Caneca 1360	Brazil
12	Seattle	2004 Charade Rd	United States of America
13	Singapore	198 Clementi North	Singapore
14	South Brunswick	2007 Zagora St	United States of America
15	South San Francisco	2011 Interiors Blvd	United States of America
16	Southlake	2014 Jabberwocky Rd	United States of America
17	Stretford	9702 Chester Road	United Kingdom
18	Sydney	12-98 Victoria Street	Australia

9. Pada tabel departments, ganti tipe data locations\_id menjadi NUMERIC(4). Kemudian tambahkan primar key pada department\_id dan foegin key pada loction\_id pada tabel departments. Alter Pertama untuk mengubah tipe Data lalu dilanjutkan dengan penambahan Constraint Primary Key dan Foreign Key

```
2430
2431 -- No 9
2432
2433 ALTER TABLE departments
2434 ALTER COLUMN location_id NUMERIC(4)
2435 GO
2436 ALTER TABLE departments
2437 ADD CONSTRAINT dept_id_pk PRIMARY KEY (department_id),
2438 CONSTRAINT dept_loc_fk FOREIGN KEY (location_id) REFERENCES locations(location_id)
```

Messages

Command(s) completed successfully.

10. Tampilkan street\_address dan postal\_code pada tabel locations serta department\_name pada tabel departments, tapi tampilkan postal\_code yang memiliki no 98199 saja. Dengan menggunakan JOIN untuk menggabungkan data dan menggunakan WHERE untuk filter postal\_code



2414	-- No 10
2415	
2416	SELECT street_address, postal_code, department_name
2417	FROM locations A JOIN departments B ON A.location_id = B.location_id
2418	WHERE A.postal_code = '98199'

Results		Messages	
	street_address	postal_code	department_name
1	2004 Charade Rd	98199	Administration
2	2004 Charade Rd	98199	Purchasing
3	2004 Charade Rd	98199	Executive
4	2004 Charade Rd	98199	Finance
5	2004 Charade Rd	98199	Accounting
6	2004 Charade Rd	98199	Treasury
7	2004 Charade Rd	98199	Corporate Tax
8	2004 Charade Rd	98199	Control And Credit
9	2004 Charade Rd	98199	Shareholder Services
10	2004 Charade Rd	98199	Benefits
11	2004 Charade Rd	98199	Manufacturing
12	2004 Charade Rd	98199	Construction
13	2004 Charade Rd	98199	Contracting
14	2004 Charade Rd	98199	Operations
15	2004 Charade Rd	98199	IT Support
16	2004 Charade Rd	98199	NOC
17	2004 Charade Rd	98199	IT Helpdesk
18	2004 Charade Rd	98199	Government Sales
19	2004 Charade Rd	98199	Retail Sales

11. Tampilkan job title, min salary, max salary pada tabel jobs, Serta first name dan salary pada tabel employee, dengan salary lebih kecil sama dengan 6000. Dengan menggunakan JOIN untuk menggabungkan data dan WHERE untuk filter salary

```

2420 -- No 11
2421
2422 SELECT job_title, min_salary, max_salary, first_name, salary
2423 FROM jobs A JOIN employees B ON A.job_id = B.job_id
2424 WHERE salary <= 6000

```

Results		Messages			
	job_title	min_salary	max_salary	first_name	salary
1	Programmer	4000	10000	Bruce	6000.00
2	Programmer	4000	10000	David	4800.00
3	Programmer	4000	10000	Valli	4800.00
4	Programmer	4000	10000	Diana	4200.00
5	Purchasing Clerk	2500	5500	Alexander	3100.00
6	Purchasing Clerk	2500	5500	Shelli	2900.00
7	Purchasing Clerk	2500	5500	Sigal	2800.00
8	Purchasing Clerk	2500	5500	Guy	2600.00
9	Purchasing Clerk	2500	5500	Karen	2500.00
10	Stock Manager	5500	8500	Kevin	5800.00
11	Stock Clerk	2000	5000	Julia	3200.00
12	Stock Clerk	2000	5000	Irene	2700.00
13	Stock Clerk	2000	5000	James	2400.00
14	Stock Clerk	2000	5000	Steven	2200.00
15	Stock Clerk	2000	5000	Laura	3300.00
16	Stock Clerk	2000	5000	Mozhe	2800.00
17	Stock Clerk	2000	5000	James	2500.00
18	Stock Clerk	2000	5000	TJ	2100.00
19	Stock Clerk	2000	5000	Jason	3300.00
20	Stock Clerk	2000	5000	Michael	2900.00
21	Stock Clerk	2000	5000	Ki	2400.00
22	Stock Clerk	2000	5000	Hazel	2200.00
23	Stock Clerk	2000	5000	Renske	3600.00

12. Tampilkan nama belakang, gaji bulanan pegawai dan kode departemen tempat pegawai tersebut bekerja khusus untuk pegawai yang bekerja pada departemen dengan kode 100 atau 101, dan memiliki gaji lebih besar atau sama dengan 1000. Dengan menggunakan JOIN untuk menggabungkan data dan menggunakan WHERE IN IN dapat digunakan untuk kondisi yang banyak

```

2426 -- No 12
2427
2428 SELECT last_name, salary, C.department_id
2429 FROM employees A JOIN jobs B ON A.job_id = B.job_id
2430 JOIN departments C ON C.department_id = A.department_id
2431 WHERE c.department_id IN (100, 101) AND salary >= 1000

```

	last_name	salary	department_id
1	Greenberg	12000.00	100
2	Faviet	9000.00	100
3	Chen	8200.00	100
4	Sciarra	7700.00	100
5	Uman	7800.00	100
6	Popp	6900.00	100

13. Tampilkan job title, min salary pada tabel jobs dan first name, salary pada tabel employees, serta department name pada tabel departments yang memiliki first name berawalan huruf "S". Dengan menggunakan JOIN untuk menggabungkan data dan juga menggunakan WHERE LIKE untuk mendapatkan data employee dengan nama depan S

```

2433 -- No 13
2434
2435 SELECT job_title, min_salary, first_name, salary, department_name
2436 FROM jobs A JOIN employees B ON A.job_id = B.job_id
2437 JOIN departments C ON C.department_id = B.department_id
2438 WHERE first_name LIKE 'S%'

```

Results Messages

	job_title	min_salary	first_name	salary	department_name
1	President	20000	Steven	24000.00	Executive
2	Accounting Manager	8200	Shelley	12000.00	Accounting
3	Sales Representative	6000	Sarath	7000.00	Sales
4	Sales Representative	6000	Sundar	6400.00	Sales
5	Sales Representative	6000	Sundita	6100.00	Sales
6	Purchasing Clerk	2500	Shelli	2900.00	Purchasing
7	Purchasing Clerk	2500	Sigal	2800.00	Purchasing
8	Stock Manager	5500	Shanta	6500.00	Shipping
9	Stock Clerk	2000	Steven	2200.00	Shipping
10	Stock Clerk	2000	Stephen	3200.00	Shipping
11	Shipping Clerk	2500	Sarah	4000.00	Shipping
12	Shipping Clerk	2500	Samuel	3200.00	Shipping
13	Human Resources Representative	4000	Susan	6500.00	Human Resources

14. Tampilkan last name, hire date, commission pct pada tabel employee, department name pada tabel department, serta city, state provincy dan country name pada tabel countries. Tampilkan commission pct NOT NULL. Dengan menggunakan JOIN dan WHERE IS NOT NULL disini kita dapat filter kolom yang valuenya tidak NULL

```

2440 -- No 14
2441
2442 SELECT last_name, hire_date, commission_pct, department_name, city, state_province, country_name
2443 FROM employees A JOIN departments B ON A.department_id = B.department_id
2444 JOIN locations C ON C.location_id = B.location_id
2445 JOIN countries D ON D.country_id = C.country_id
2446 WHERE commission_pct IS NOT NULL

```

	last_name	hire_date	commission_pct	department_name	city	state_province	country_name
1	Russell	1996-10-01	0.40	Sales	Oxford	Oxford	United Kingdom
2	Partners	1997-01-05	0.30	Sales	Oxford	Oxford	United Kingdom
3	Errazuriz	1997-03-10	0.30	Sales	Oxford	Oxford	United Kingdom
4	Cambrault	1999-10-15	0.30	Sales	Oxford	Oxford	United Kingdom
5	Zlotkey	2000-01-29	0.20	Sales	Oxford	Oxford	United Kingdom
6	Tucker	1997-01-30	0.30	Sales	Oxford	Oxford	United Kingdom
7	Bernstein	1997-03-24	0.25	Sales	Oxford	Oxford	United Kingdom
8	Hall	1997-08-20	0.25	Sales	Oxford	Oxford	United Kingdom
9	Olsen	1998-03-30	0.20	Sales	Oxford	Oxford	United Kingdom
10	Cambrault	1998-12-09	0.20	Sales	Oxford	Oxford	United Kingdom
11	Tuvault	1999-11-23	0.15	Sales	Oxford	Oxford	United Kingdom
12	King	1996-01-30	0.35	Sales	Oxford	Oxford	United Kingdom
13	Sully	1996-03-04	0.35	Sales	Oxford	Oxford	United Kingdom
14	McEwen	1996-08-01	0.35	Sales	Oxford	Oxford	United Kingdom
15	Smith	1997-03-10	0.30	Sales	Oxford	Oxford	United Kingdom
16	Doran	1997-12-15	0.30	Sales	Oxford	Oxford	United Kingdom
17	Sewall	1998-11-03	0.25	Sales	Oxford	Oxford	United Kingdom
18	Vishney	1997-11-11	0.25	Sales	Oxford	Oxford	United Kingdom
19	Greene	1999-03-19	0.15	Sales	Oxford	Oxford	United Kingdom
20	Marvins	2000-01-24	0.10	Sales	Oxford	Oxford	United Kingdom
21	Lee	2000-02-23	0.10	Sales	Oxford	Oxford	United Kingdom
22	Ande	2000-03-24	0.10	Sales	Oxford	Oxford	United Kingdom
23	Banda	2000-04-21	0.10	Sales	Oxford	Oxford	United Kingdom
24	Ozer	1997-03-11	0.25	Sales	Oxford	Oxford	United Kingdom

# **PENUTUP**

## **1. Kesimpulan**

Pada praktikum ini telah mempelajari DML atau Data Manipulation Language seperti SELECT, INSERT, UPDATE, DELETE dan juga Cara membuat Table, Join Table serta Constraint pada Table selain itu juga mempraktekan DML dengan Database lepkom dan menampilkan data sesuai dengan soal yang diminta.

## **2. SaranPenulisan**





# **LAPORAN AKHIR PROYEK**



STORED PROCEDURE DAN FUNCTION

Disusun Oleh:

Nama : 11117320

NPM : Chantika Amanda

Kelas :

**LEMBAGA PENGEMBANGAN KOMPUTERISASI  
UNIVERSITAS GUNADARMA**

**2022**

# DAFTAR ISI

DESKRIPSI LAPORAN .....	3
TAHAPAN Pengerjaan.....	4
PENUTUP .....	9



# **DESKRIPSI LAPORAN**

## **STORED PROCEDURE DAN FUNCTION**

Pada pertemuan 3 laporan berisikan cara membuat Stored Procedure dan juga Function, Jenis jenis Stored Procedure dan Function, serta cara melakukan modifikasi seperti membuat, mengubah dan melakukan delete



# TAHAPAN Pengerjaan

## 1. Ringkasan Materi

Stored procedure merupakan objek pre-compile yang di-compile satu kali dan format yang di-compile kemudian disimpan dan format tersebut akan mengeksekusi kode yang di-compile setiap kali procedure tersebut dipanggil. Sedangkan function di-compile dan dieksekusi setiap kali function tersebut dipanggil.

Function	Stored Procedure
Harus mengembalikan nilai	Tidak harus mengembalikan nilai
Hanya dapat memiliki input parameter	Dapat memiliki parameter output input dan
Tidak dapat memanggil procedure	Dapat memanggil function
Hanya boleh terdapat SELECT statement	Boleh terdapat statement statement DML lainnya select dan
Tidak boleh terdapat blok try-catch	Boleh menggunakan blok try-catch untuk penanganan error
Tidak boleh terdapat transaction	Boleh menggunakan transaction
Dapat dipanggil dalam SELECT statement	Tidak dapat dipanggil dalam statement SELECT/WHERE/HAVING, hanya dapat digunakan dengan statement EXECUTE/EXEC
Dapat digunakan dalam JOIN sebagai result set	Tidak dapat digunakan dalam klausa JOIN

Parameter digunakan untuk pertukaran data antara stored procedure dan function dengan suatu aplikasi atau alat bantu yang memanggil stored procedure atau function tersebut. Terdapat beberapa jenis parameter yaitu sebagai berikut.

- Parameter input, memungkinkan pemanggil untuk memberikan nilai data ke stored procedure atau function.
- Parameter output, memungkinkan stored procedure untuk memberikan nilai data atau variable cursor kembali ke pemanggil. Function tidak dapat menspesifikasi parameter output.

- Return code, merupakan nilai integer yang dikembalikan oleh stored procedure ke pemanggil. Jika stored procedure tidak mengatur nilai untuk return code secara explicit, maka return code otomatis akan bernilai 0.

Stored procedure dalam SQL Server adalah kumpulan dari satu atau lebih pernyataan Transact-SQL atau suatu referensi ke metode Common Runtime Language (CLR) Framework Microsoft .NET. Procedure dapat dikatakan seperti construct pada bahasa pemrograman lainnya karena procedure dapat melakukan beberapa hal seperti berikut.

- Menerima parameter input dan mengembalikan banyak nilai dalam bentuk parameter output ke program yang dipanggil.
- Mengandung pernyataan pemrograman yang dapat melakukan berbagai operasi dalam database, termasuk memanggil procedure lainnya.
- Mengembalikan nilai status ke program yang dipanggil untuk mengindikasikan kesuksesan atau kegagalannya, beserta penyebab kegagalan tersebut.

#### Manfaat Menggunakan Stored Procedure

- Mengurangi Traffic Jaringan Server/Client Perintah pada suatu procedure dieksekusi sebagai satu batch kode. Hal ini dapat mengurangi traffic jaringan secara signifikan antara server dan klien karena hanya panggilan untuk mengeksekusi procedure yang dikirim melalui jaringan. Tanpa enkapsulasi kode yang disediakan oleh suatu procedure, setiap baris individual dari kode harus dikirim melalui jaringan.
- Memperkuat Keamanan Banyak user dan program klien dapat melakukan operasi pada objek- objek database pokok melalui suatu procedure, meskipun user dan program tersebut tidak memiliki hak akses langsung pada objek-objek pokok tersebut. Procedure tersebut mengontrol proses dan aktifitas apa yang dapat dilakukan, serta tetap melindungi objek-objek database pokok tersebut. Hal ini dapat mengurangi kebutuhan untuk memberikan hak akses pada level objek individual dan menyederhanakan layer keamanan
- Penggunaan Kode Secara Berulang Kode untuk setiap operasi database yang berulang adalah kandidat sempurna untuk enkapsulasi pada procedure. Hal ini dapat mengurangi kebutuhan untuk menulis ulang kode yang sama, mengurangi ketidakkonsistenan kode yang dibuat, dan memungkinkan suatu kode dapat diakses dan dieksekusi oleh semua user atau aplikasi yang memproses hak akses yang dibutuhkan.
- Mempermudah Pemeliharaan Ketika aplikasi klien memanggil procedure dan menyimpan operasi database pada tingkat data, hanya procedure yang harus di-update untuk setiap perubahan pada database pokok. Level aplikasi tetap terpisah dan tidak perlu mengetahui perubahan-perubahan pada layout, relationship, atau proses pada database.

- Meningkatkan Performance Secara default, suatu procedure telah ter-compile pada saat pertama kali dieksekusi dan membuat suatu rencana eksekusi yang digunakan Kembali untuk eksekusi selanjutnya. Karena prosesor query tidak harus membuat rencana baru, secara tipikal akan mengurangi waktu untuk memproses procedure. Jika terdapat perubahan signifikan pada table atau data yang direferensikan oleh procedure, rencana yang telah di-compile sebelumnya dapat menyebabkan procedure berjalan lebih lambat. Dalam hal ini, melakukan compile ulang pada procedure dan memaksa pembuatan rencana eksekusi baru dapat meningkatkan performance.

Jenis- Jenis Stored Procedure :

- User-Defined
- Temporary
- System
- Extended User-Defined

Function Sama seperti function pada bahasa pemrograman, user-defined function pada SQL Server adalah suatu rutin yang dapat menerima parameter, melakukan suatu aksi seperti perhitungan kompleks, dan mengembalikan hasil dari aksi tersebut sebagai suatu nilai. Nilai yang dikembalikan dapat berupa satu nilai skalar maupun result set

Manfaat Menggunakan User-Defined Function

- Memungkinkan pemrograman modular Suatu function dapat dibuat satu kali, disimpan di database, dan dipanggil berapa kali pun dalam suatu program. Function user-defined dapat diubah secara terpisah dari kode sumber program.
- Mempercepat eksekusi Mirip seperti stored procedure, function user-defined Transact-SQL mengurangi biaya kompilasi dari kode Transact-SQL dengan menyembunyikan rencana dan menggunakannya kembali untuk dieksekusi berulang kali. Hal ini berarti function user-defined tidak perlu diurai atau dioptimasi ulang dengan setiap penggunaan sehingga akan membuat waktu eksekusi menjadi lebih cepat.
- Mengurangi traffic jaringan Suatu operasi yang menyaring data berdasarkan beberapa constraint kompleks yang tidak dapat diekspresikan pada satu ekspresi skalar dapat diekspresikan dalam suatu function. Function tersebut kemudian dapat diminta pada klausa WHERE untuk mengurangi jumlah baris yang dikirim ke klien

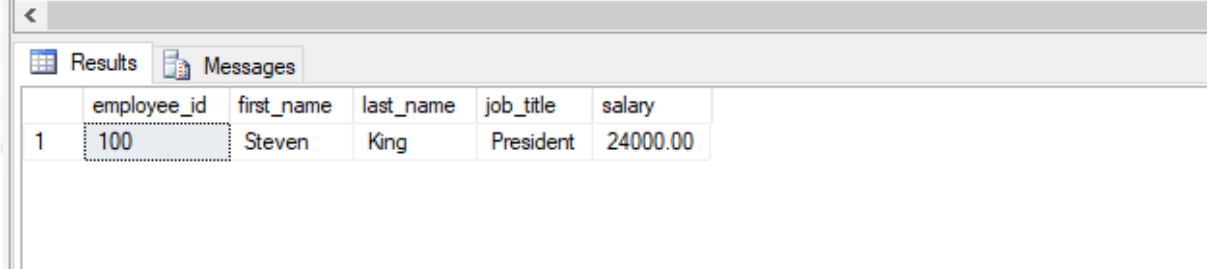
Jenis Jenis Function :

- Scalar Function
- Table-Valued Functions
- System Functions

## 2. Langkah-Langkah

1. Buat Stored Procedure mencari Karyawan. Dengan isi procedure melakukan Select terhadap employee\_id, first\_name, last\_name, job\_title, dan salary di table employees yang di join dengan table jobs dan difilter employee\_id sesuai dengan parameter

```
2448
2449 -- Tugas 1
2450
2451 GO
2452 CREATE PROCEDURE cariKaryawan_11117320 @employee_id NUMERIC(6)
2453 AS
2454 BEGIN
2455     SELECT employee_id, first_name, last_name, job_title, salary
2456     FROM employees A JOIN jobs B ON A.job_id = B.job_id
2457     WHERE employee_id = @employee_id
2458 END
2459 GO
2460
2461 EXEC cariKaryawan_11117320 100
2462
2463
2464 -- Tugas 1 Part 2
```



The screenshot shows the SQL Server Enterprise Manager interface. The top pane displays the SQL script for creating and executing a stored procedure. The bottom pane shows the results of the execution, which is a table with one row of data.

	employee_id	first_name	last_name	job_title	salary
1	100	Steven	King	President	24000.00

2. Buat Stored Procedure Ubah Salary Karyawan. Yang berisikan statement update table employees dan set salarynya menjadi salary di parameter dimana employee\_id nya sama dengan employee\_id pada parameter

```
2463
2464 -- Tugas 1 Part 2
2465
2466 GO
2467 CREATE PROCEDURE ubahSalary_11117320 @employee_id NUMERIC(6), @salary NUMERIC(8,2)
2468 AS
2469 BEGIN
2470     UPDATE employees SET salary = @salary WHERE employee_id = @employee_id
2471 END
2472 GO
2473
2474 EXEC ubahSalary_11117320 100, 25000.00
```

Messages

(1 row(s) affected)





# **PENUTUP**

## **1. Kesimpulan**

Pada praktikum ini telah mempelajari mengenai apa itu Stored Procedure dan Function pada SQL Serta mempelajari cara membuat dan melakukan modifikasi Stored Procedure dan Function, mempelajari parameter di Stored Procedure dan Function serta membuat Stored Procedure pada bagian Tugas.

## **2. SaranPenulisan**



# **LAPORAN AKHIR PROYEK**



**LOGIN – USER – ROLE - PERMISSION**

**Disusun Oleh:**

**Nama : 11117320**

**NPM : Chantika Amanda**

**Kelas :**

**LEMBAGA PENGEMBANGAN KOMPUTERISASI  
UNIVERSITAS GUNADARMA  
2022**

# DAFTAR ISI

DESKRIPSI LAPORAN .....	3
TAHAPAN Pengerjaan.....	4
PENUTUP .....	10



# **DESKRIPSI LAPORAN**

## **LOGIN – USER – ROLE - PERMISSION**

Pada pertemuan 4 laporan berisikan cara membuat login, user, role dan juga permission selain itu juga mengenai cara memodifikasi login, user, role dan juga permission



# TAHAPAN Pengerjaan

## 1. Ringkasan Materi

Login merupakan kepala dari suatu sistem keamanan atau sebuah entitas yang dapat di konfirmasi oleh sistem keamanan. Setiap pengguna harus melakukan login untuk bisa terkoneksi ke dalam SQL server

User adalah bagian dari database level security. Login harus terhubung dengan database user untuk saling terhubung. Login dapat terhubung dengan database yang berbeda sebagai pengguna yang berbeda tapi bisa juga hanya terhubung dengan satu pengguna pada masing-masing database. User dapat dibuat tanpa harus memiliki login

Gunakan role didalam SQL Server untuk mengelola permission secara efisien. Memberikan permission untuk role, kemudian menambah dan menghapus user serta login untuk role. Dengan menggunakan role, permission tidak harus di atur secara individual untuk setiap pengguna.

SQL Server mendukung 4 tipe role, yaitu :

- Fixed server roles
- User-defined server roles
- Fixed database roles
- User- defined database roles

Berikut adalah jenis-jenis role untuk fixed server-level role :

FIXED SERVER-LEVEL ROLE	DESCRIPTION
<b>sysadmin</b>	Members of the <b>sysadmin</b> fixed server role can perform any activity in the server.
<b>serveradmin</b>	Members of the <b>serveradmin</b> fixed server role can change server-wide configuration options and shut down the server.
<b>securityadmin</b>	<p>Members of the <b>securityadmin</b> fixed server role manage logins and their properties. They can <b>GRANT</b> , <b>DENY</b> , and <b>REVOKE</b> server-level permissions. They can also <b>GRANT</b> , <b>DENY</b> , and <b>REVOKE</b> database-level permissions if they have access to a database. Additionally, they can reset passwords for SQL Server logins.</p> <p><b>IMPORTANT:</b> The ability to grant access to the Database Engine and to configure user permissions allows the security admin to assign most server permissions. The <b>securityadmin</b> role should be treated as equivalent to the <b>sysadmin</b> role.</p>
<b>processadmin</b>	Members of the <b>processadmin</b> fixed server role can end processes that are running in an instance of SQL Server.
<b>setupadmin</b>	Members of the <b>setupadmin</b> fixed server role can add and remove linked servers by using Transact-SQL statements. ( <b>sysadmin</b> membership is needed when using Management Studio.)
<b>bulkadmin</b>	Members of the <b>bulkadmin</b> fixed server role can run the <b>BULK INSERT</b> statement.
<b>diskadmin</b>	The <b>diskadmin</b> fixed server role is used for managing disk files.
<b>dbcreator</b>	Members of the <b>dbcreator</b> fixed server role can create, alter, drop, and restore any database.
<b>public</b>	<p>Every SQL Server login belongs to the <b>public</b> server role. When a server principal has not been granted or denied specific permissions on a securable object, the user inherits the permissions granted to public on that object. Only assign public permissions on any object when you want the object to be available to all users. You cannot change membership in public.</p> <p><b>Note:</b> <b>public</b> is implemented differently than other roles, and permissions can be granted, denied, or revoked from the public fixed server roles.</p>

Berikut adalah jenis-jenis role untuk fixed database roles :

FIXED-DATABASE ROLE NAME	DESCRIPTION
<b>db_owner</b>	Members of the <b>db_owner</b> fixed database role can perform all configuration and maintenance activities on the database, and can also drop the database in SQL Server. (In SQL Database and SQL Data Warehouse, some maintenance activities require server-level permissions and cannot be performed by <b>db_owners</b> .)
<b>db_securityadmin</b>	Members of the <b>db_securityadmin</b> fixed database role can modify role membership and manage permissions. Adding principals to this role could enable unintended privilege escalation.
<b>db_accessadmin</b>	Members of the <b>db_accessadmin</b> fixed database role can add or remove access to the database for Windows logins, Windows groups, and SQL Server logins.
<b>db_backupoperator</b>	Members of the <b>db_backupoperator</b> fixed database role can back up the database.
<b>db_ddladmin</b>	Members of the <b>db_ddladmin</b> fixed database role can run any Data Definition Language (DDL) command in a database.
<b>db_datawriter</b>	Members of the <b>db_datawriter</b> fixed database role can add, delete, or change data in all user tables.
<b>db_datareader</b>	Members of the <b>db_datareader</b> fixed database role can read all data from all user tables.
<b>db_denydatawriter</b>	Members of the <b>db_denydatawriter</b> fixed database role cannot add, modify, or delete any data in the user tables within a database.
<b>db_denydatareader</b>	Members of the <b>db_denydatareader</b> fixed database role cannot read any data in the user tables within a database.

Berikut adalah jenis-jenis role untuk user defined database roles :

FEATURE	TYPE	DESCRIPTION
<a href="#">sp_helpdbfixedrole</a> (Transact-SQL)	Metadata	Returns a list of the fixed database roles.
<a href="#">sp_dbfixedrolepermission</a> (Transact-SQL)	Metadata	Displays the permissions of a fixed database role.
<a href="#">sp_helprole</a> (Transact-SQL)	Metadata	Returns information about the roles in the current database.
<a href="#">sp_helprolemember</a> (Transact-SQL)	Metadata	Returns information about the members of a role in the current database.
<a href="#">sys.database_role_members</a> (Transact-SQL)	Metadata	Returns one row for each member of each database role.
<a href="#">IS_MEMBER</a> (Transact-SQL)	Metadata	Indicates whether the current user is a member of the specified Microsoft Windows group or Microsoft SQL Server database role.
<a href="#">CREATE ROLE</a> (Transact-SQL)	Command	Creates a new database role in the current database.
<a href="#">ALTER ROLE</a> (Transact-SQL)	Command	Changes the name or membership of a database role.
<a href="#">DROP ROLE</a> (Transact-SQL)	Command	Removes a role from the database.
<a href="#">sp_addrole</a> (Transact-SQL)	Command	Creates a new database role in the current database.
<a href="#">sp_droprole</a> (Transact-SQL)	Command	Removes a database role from the current database.
<a href="#">sp_addrolemember</a> (Transact-SQL)	Command	Adds a database user, database role, Windows login, or Windows group to a database role in the current database. All platforms except Parallel Data Warehouse should use <code>ALTER ROLE</code> instead.
<a href="#">sp_droprolemember</a> (Transact-SQL)	Command	Removes a security account from a SQL Server role in the current database. All platforms except Parallel Data Warehouse should use <code>ALTER ROLE</code> instead.
<a href="#">GRANT</a>	Permissions	Adds permission to a role.
<a href="#">DENY</a>	Permissions	Denys a permission to a role.
<a href="#">REVOKE</a>	Permissions	Removes a previously granted or denied permissions.

Owner basis data, member dari sysadmin, dan member dari securityadmin dapat memberikan akses basis data, dan hak akses yang diberikan meliputi :



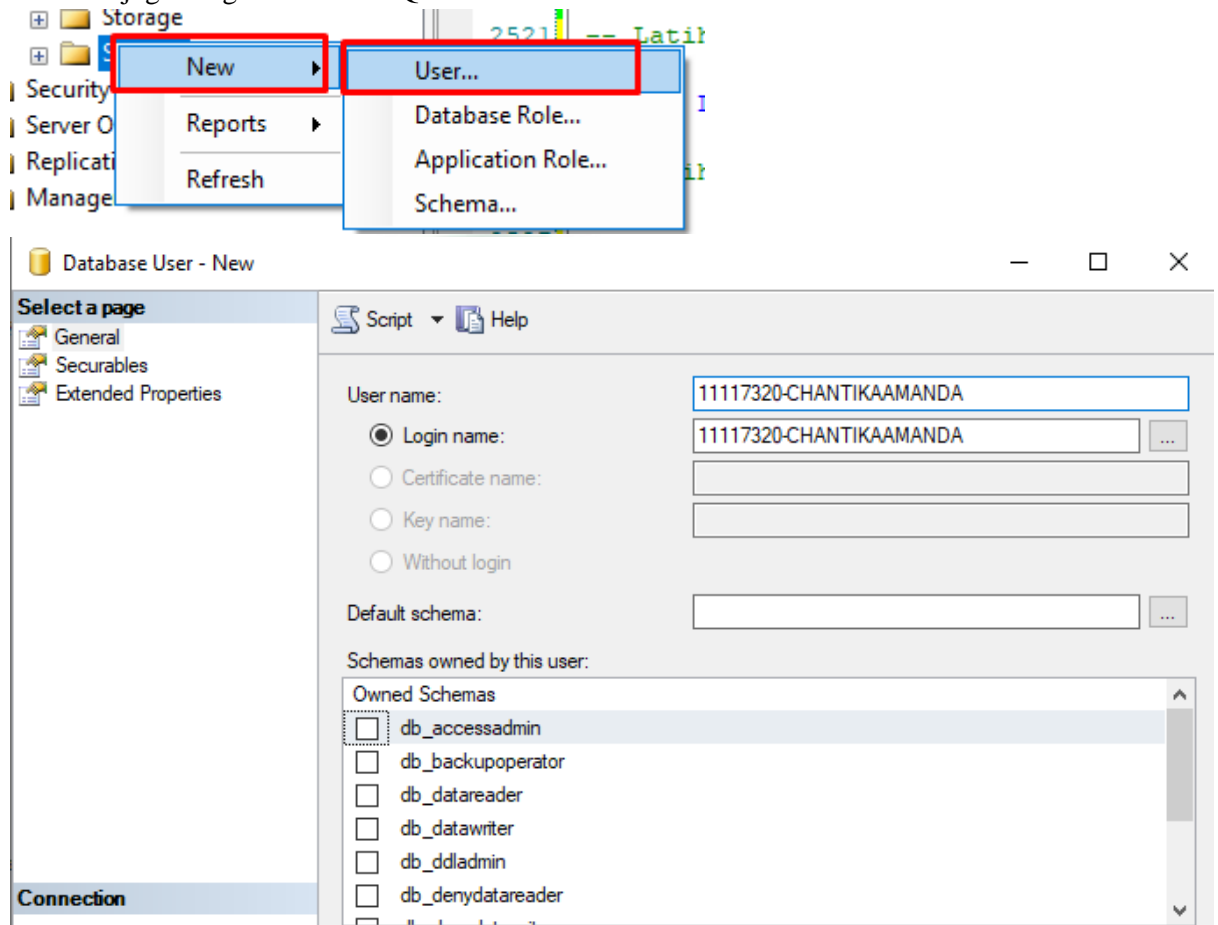
- Grant Grant merupakan perintah yang digunakan untuk memberikan hak akses. Dengan role, semua member dari role akan mendapatkan hak akses
- Revoke Revoke merupakan perintah yang digunakan untuk menghapus hak akses. Menghilangkan hak akses GRANT.
- Deny

## 2. Langkah-Langkah

1. Buat Login dengan nama CHANTIKAAMANDA dan Password NPM

```
CREATE LOGIN CHANTIKAAMANDA WITH PASSWORD = '11117320';
```

2. Buat User dengan melalui SSMS Click kanan pada Security dalam Database lalu New -> User dan juga dengan Transact SQL lalu buat table BIODATA



```
CREATE USER CHANTIKAAMANDA_11117320 FOR LOGIN CHANTIKAAMANDA;
```

```
529 -- Table BioData
530
531 CREATE TABLE BIODATA (
532     NPM NUMERIC(8) PRIMARY KEY,
533     NAMA VARCHAR(50) NOT NULL,
534     KELAS VARCHAR(5) NOT NULL,
535     JURUSAN VARCHAR(30) NOT NULL
536 )
```

3. Buat Role dengan Nama dan Role securityadmin lalu berikan access SELECT, INSERT, UPDATE, DELETE pada table BIODATA yang dibuat sebelumnya  
**Sebutkan empat macam jenis-jenis role yang biasa digunakan dalam basis data di SQL Server**

SQL Server mendukung 4 tipe role, yaitu :

1. Fixed server roles
2. User-defined server roles
3. Fixed database roles
4. User-defined database roles

```
-- Latihan Role

CREATE ROLE chantikamanda AUTHORIZATION securityadmin
-- Grant Permission

GRANT SELECT, INSERT, UPDATE, DELETE
ON SCHEMA::BIODATA
TO chantikamanda;

INSERT INTO BIODATA VALUES ('12345678', 'Ani Budi Citra', '2KA05', 'Sistem Informasi')
INSERT INTO BIODATA VALUES ('56789012', 'Dea Efani Fita Gita', '1IA01', 'Teknik Informatika')
INSERT INTO BIODATA VALUES ('34567890', 'Hari Irfandi Jonas', '3DB04', 'Manajemen Informatika')
INSERT INTO BIODATA VALUES ('23456789', 'Kamia Lana Musa', '4KB02', 'Sistem Komputer')
INSERT INTO BIODATA VALUES ('45678901', 'Nando Oprah Prawira', '2DC03', 'Teknik Komputer')

-- Tugas SQL Injection
```

Messages

(1 row(s) affected)

(1 row(s) affected)

(1 row(s) affected)

(1 row(s) affected)

(1 row(s) affected)

# **PENUTUP**

## **1. Kesimpulan**

Pada praktikum ini telah mempelajari mengenai Login, User, Role, Permission dan juga mempraktekannya dengan membuat Login, User, Role dan melakukan Set Permission

## **2. SaranPenulisan**



# **LAPORAN AKHIR PROYEK**



SQL SERVER SECURITY

Disusun Oleh:

Nama : 11117320

NPM : Chantika Amanda

Kelas :

**LEMBAGA PENGEMBANGAN KOMPUTERISASI  
UNIVERSITAS GUNADARMA**

**2022**

# DAFTAR ISI

DESKRIPSI LAPORAN .....	3
TAHAPAN Pengerjaan.....	4
PENUTUP .....	9



# **DESKRIPSI LAPORAN**

## **SQL SERVER SECURITY**

Pada pertemuan 5 laporan berisikan mengenai Security Principals dan Securables di SQL Server, memahami Mode otentikasi di SQL Server, memahami users dan logins di SQL Server, dapat menerapkan password policy, dan mencegah SQL Injection



# TAHAPAN Pengerjaan

## 1. Ringkasan Materi

Entitas yang dapat meminta/ memiliki akses ke server, database, atau schema disebut sebagai security principals. Sedangkan item yang dapat diamankan untuk mengontrol akses terhadap item tersebut disebut securables. Security principals dapat digolongkan menjadi 2 kelompok yaitu:

- Indivisible principals adalah sebuah entitas yang tidak bergantung terhadap entitas lainnya. Contohnya yaitu SQL login dan Windows login.
- Collection principals adalah principal yang merepresentasikan sekelompok entitas yang diperlakukan sebagai sebuah kesatuan. Contoh dari principal ini yaitu Windows group.

Terdapat tiga level keamanan yang dapat diolah pada security principals SQL Server, yaitu Windows, SQL Server, dan database. Pada masing-masing level terdapat lagi security principals yang dapat diolah. Berikut ini principals yang terdapat pada masing-masing tingkatan.

- Windows level principals
  - Domain logins
  - Local system logins
- SQL Server level principals
  - SQL Server login
  - Windows login
- Database level principals
  - Users
  - Roles
  - Application roles

Securables adalah semua resources dalam SQL Server yang dapat diatur hak aksesnya. Contoh securable yaitu sebuah tabel. Terdapat tiga tingkat cakupan dari securables yaitu server, database, dan schema. Berikut ini securables pada masing-masing tingkatan.

- Server
  - SQL Server instance
  - Endpoint
  - Login
  - Server role
  - Database
- Database
  - Application role

- Assembly
- Asymmetric key
- Certificate
- Contract
- Fulltext catalog
- Fulltextstoplist
- Message type
- Remote Service Binding
- (Database) Role
- Route
- Schema
- Search property list
- Service
- Symmetric key
- User
- Schema
  - Type
  - XML schema collection
  - Object. Berikut ini yang termasuk ke dalam object pada schema:
    - Aggregate
    - Function
    - Procedure
    - Queue
    - Synonym
    - Table
    - View
    - External Table

Terdapat dua macam mode otentikasi pada SQL Server yaitu:

- Mode otentikasi Windows: Penggunaan mode ini paling cocok ketika database hanya diakses dalam satu lingkup organisasi.
- Mode otentikasi SQL Server dan Windows (mixed mode): Penggunaan mode ini paling cocok ketika database juga diakses oleh pengguna yang berada di luar lingkup suatu organisasi (diluar domain Windows) atau pun pengguna yang tidak menggunakan perangkat Windows.

Dengan menggunakan mode otentikasi Windows, pengguna dapat login ke SQL Server baik dengan menggunakan user account maupun group account. Mode otentikasi Windows ini lebih mudah user untuk masuk ke dalam SQL Server karena user tidak perlu untuk mengingat username SQL Server login dan password- nya.

Dengan keamanan campuran (mixed security), pengguna menggunakan otentikasi Windows dan login SQL Server. Login SQL Server terutama untuk pengguna di luar perusahaan, seperti mereka yang mungkin mengakses database dari Internet. Anda dapat



mengkonfigurasi aplikasi yang mengakses SQL Server dari Page 5 Internet untuk menggunakan akun tertentu secara otomatis atau untuk meminta pengguna memasukkan ID masuk dan kata sandi SQL Server.

Sebagaimana yang telah dibahas sebelumnya bahwa SQL Server memiliki 2 metode otentikasi yang berarti terdapat 2 jenis login server. Pengguna bisa login Windows dengan menggunakan domain account atau local account, local group account atau universal dan global domain group account. Pengguna bisa juga menggunakan login SQL Server dengan menentukan ID login dan password yang unik. Beberapa login dikonfigurasi secara default, yaitu local Administrators, local Administrator, sa. Penggunaan user untuk mempersempit ruang lingkup akses ke database. Beberapa users dikonfigurasi secara default, termasuk the dbo user, dan the guest user

### **Administrators Group**

Administrators group adalah sebuah grup lokal dalam server database. Anggota dari grup ini biasanya akun local Administrator user dan user lainnya yang telah diatur sebagai administrator dalam lokal sistem. Di SQL Server, grup ini diberikan hak akses sebagai sysadmin server role secara default.

### **Administrator User Account**

Administrator adalah sebuah user account lokal di server. Akun ini memberikan hak akses sebagai administrator pada sistem. Jika SQL Server terpasang di Windows domain, maka administrator account biasanya memiliki hak akses secara domain juga. Pada SQL Server, akun ini mendapatkan hak akses sysadmin server role secara default.

### **Sa Login**

Sa login adalah akun sistem administrator di SQL Server. Dengan model keamanan baru yang telah terintegrasi dan diperluas, sa tidak lagi dibutuhkan. Sa disediakan untuk kompatibilitas terhadap SQL Server versi sebelumnya. Seperti login administrator lainnya, sa diberikan server sysadmin role secara default. Saat Anda menginstal SQL Server, login sa tidak diberi kata sandi. Untuk mencegah akses yang tidak sah ke server, kata sandi harus benar-benar kuat, dan harus diganti secara berkala.

### **Guest User**

Guest User adalah pengguna khusus yang dapat Anda tambahkan ke database untuk memungkinkan seseorang dengan login SQL Server yang valid untuk mengakses database. Pengguna yang mengakses database dengan guest account menganggap identitas pengguna tamu dan mewarisi semua hak istimewa dan hak akses akun tamu. Misalnya, jika Anda mengonfigurasi akun GOTEAM untuk mengakses SQL Server, GOTEAM dapat mengakses database apa pun dengan info masuk tamu, dan saat GOTEAM melakukannya, orang yang masuk di bawah GOTEAM diberikan semua hak akses akun tamu. Jika Anda mengkonfigurasi grup Windows DEVGROUP dengan akses tamu, Anda dapat menyederhanakan administrasi karena setiap pengguna yang merupakan anggota grup akan

dapat mengakses database apa pun sebagai tamu. Guest user adalah user khusus yang dapat ditambahkan ke database untuk memungkinkan seseorang dengan login SQL Server yang valid untuk mengakses database. User yang mengakses database dengan guest account mewarisi semua privileges dan permission yang dimiliki oleh guest user. Secara default, guest user terdapat di database model. Karena database model adalah template untuk semua database yang dibuat, semua database baru memiliki guest user. Guest user tidak dapat dihapus namun bisa dinonaktifkan kecuali pada database master dan tempdb. Hal ini tidak menjadi masalah karena guest user memiliki privilege dan permission yang terbatas di database tersebut.

## 2. Langkah-Langkah

1. **Manakah mode otentikasi yang lebih baik antara mode Windows dan mode campuran? Sertakan alasannya!**

Mode otentikasi Campuran lebih baik karena Penggunaan mode ini dapat digunakan ketika database juga diakses oleh pengguna yang berada di luar lingkup suatu organisasi (diluar domain Windows) atau pun pengguna yang tidak menggunakan perangkat Windows Selain itu Anda dapat mengkonfigurasi aplikasi yang mengakses SQL Server dari Page 5 Internet untuk menggunakan akun tertentu secara otomatis atau untuk meminta pengguna memasukkan ID masuk dan kata sandi SQL Server

2. **Buatlah password policy untuk perusahaan x pada Local/Domain Security Policy, berikan alasan yang konkrit untuk masing-masing kebijakan/policy yang telah Anda tentukan**

Kebijakan yang akan dibuat yaitu "*enforce password history = 12*" sehingga dalam 1 tahun tidak ada password yang berulang selanjutnya "*maximum password age = 30*" yang artinya setiap 1 bulan sekali user harus melakukan perubahan password selanjutnya "*minimum password age = 15*" ini artinya password yang baru diganti tidak dapat diganti lagi dalam waktu < 15 hari selanjutnya "*minimum password length = 9*" sehingga minimal jumlah karakter untuk suatu password yaitu 9 character dengan mengikuti aturan "*Password must meet complexity requirements*" default

3. **SQL Injection** (Dengan menggunakan sp\_executesql dan param maka kita telah berhasil mencegah SQL Injection yang mencoba menampilkan seluruh user Login pada Instance yang sedang di Hit, selain itu fungsi dari Procedure masih berjalan baik dan dapat dilihat pada Gambar kedua)

```
Tugas Lepkom Man...vcamaster (52)
2521 -- Tugas SQL Injection
2522
2523 GO
2524 ALTER PROCEDURE pekerjaDariManager_11117320 @manager_first_name VARCHAR(MAX), @manager_last_name VARCHAR(MAX)
2525 AS
2526 BEGIN
2527
2528     DECLARE @sqlcmd NVARCHAR(MAX);
2529     DECLARE @params NVARCHAR(MAX);
2530
2531     SET @sqlcmd = N'SELECT first_name, department_name
2532 FROM employees A JOIN departments B ON A.department_id = B.department_id
2533 WHERE A.manager_id = (SELECT employee_id FROM employees WHERE first_name = @manager_first_name AND last_name = @manager_last_name)';
2534
2535     SET @params = N'@manager_first_name VARCHAR(MAX), @manager_last_name VARCHAR(MAX)';
2536
2537     EXECUTE sp_executesql @sqlcmd, @params, @manager_first_name, @manager_last_name;
2538
2539 END
2540 GO
2541
2542 DECLARE @first_name VARCHAR(MAX), @last_name VARCHAR(MAX);
2543 SET @first_name = 'Steven';
2544 SET @last_name = 'King'; UNION SELECT name, password_hash FROM master.sys.sql_logins; -- Test Inject Get User Login';
2545 EXEC pekerjaDariManager_11117320 @first_name, @last_name
```

first_name	department_name
------------	-----------------

```
2540 GO
2541
2542 DECLARE @first_name VARCHAR(MAX), @last_name VARCHAR(MAX);
2543 SET @first_name = 'Steven';
2544 SET @last_name = 'King'
2545 EXEC pekerjaDariManager_11117320 @first_name, @last_name
```

first_name	department_name
------------	-----------------

	first_name	department_name
1	Neena	Executive
2	Lex	Executive
3	Den	Purchasing
4	Matthew	Shipping
5	Adam	Shipping
6	Payam	Shipping
7	Shanta	Shipping
8	Kevin	Shipping
9	John	Sales
10	Karen	Sales
11	Alberto	Sales
12	Gerald	Sales
13	Eleni	Sales
14	Michael	Marketing

# **PENUTUP**

## **1. Kesimpulan**

Pada Praktikum ini telah mempelajari mengenai Security Principal pada SQL Server, memahami mengenai Metode Otentikasi Kelebihan dan Kekurangannya, memahami mengenai apa itu SQL Injection dan bagaimana cara mencegahnya, memahami Users dan Login pada SQL Server dan menerapkan Password Policy

## **2. SaranPenulisan**



# **LAPORAN AKHIR PROYEK**



PERFORMING DATABASE BACKUP

Disusun Oleh:

Nama : 11117320

NPM : Chantika Amanda

Kelas :

**LEMBAGA PENGEMBANGAN KOMPUTERISASI  
UNIVERSITAS GUNADARMA**

**2022**

# DAFTAR ISI

DESKRIPSI LAPORAN .....	3
TAHAPAN Pengerjaan.....	4
PENUTUP .....	12



# **DESKRIPSI LAPORAN**

## **PERFORMING DATABASE BACKUP**

Pada pertemuan 6 laporan berisikan mengenai cara Backup memilih perangkat backup , strategi backup, dan melakukan backup



# TAHAPAN Pengerjaan

## 1. Ringkasan Materi

Backup adalah bagian penting dari administrasi database. Backup sangat penting sehingga SQL Server menyediakan beberapa prosedur Backup dan beberapa cara untuk membuat backup semuanya dirancang untuk membantu Anda mengelola backup dan pemulihan database dengan mudah dan efektif

Membuat dan menerapkan rencana backup dan recovery membutuhkan waktu. Anda perlu mencari tahu database mana yang perlu dibackup, seberapa sering database harus di backup, dan detail lainnya

1. Apa jenis database yang akan dibackup?
2. Seberapa pentingkah data dalam database?
3. Seberapa sering perubahan dilakukan pada database?
4. Seberapa cepat perlu memulihkan data?
5. Apakah memiliki peralatan untuk melakukan backup?
6. Bisakah memampatkan backup?
7. Kapan waktu terbaik untuk menjadwalkan backup?
8. Apakah perlu menyimpan backup diluar lokasi saat ini?

Membuat backup database berbeda dengan membackup server atau workstation, terutama karena sering perlu menggabungkan semua (atau hampir semua) teknik yang ada untuk memastikan bahwa dapat memulihkan database sepenuhnya. Jenis backup dasar yang dapat dilakukan meliputi:

1. Backup database lengkap (Full database backup). Backup database lengkap mencakup semua objek, tabel sistem, dan data. Saat backup dimulai, SQL Server menyalin semua yang ada di database dan juga menyertakan bagian dari log transaksi yang dibutuhkan saat backup sedang berlangsung. Full backup dapat digunakan untuk memulihkan keadaan data lengkap dalam database pada saat operasi backup selesai.
2. Diferensial backup. Diferensial backup dirancang untuk membuat backup data yang telah berubah sejak full backup terakhir. Karena hanya perubahan yang tersimpan, jenis cadangan ini memerlukan sedikit waktu, dan dapat dilakukan lebih sering. Seperti backup penuh, backup diferensial termasuk bagian dari log transaksi yang diperlukan untuk mengembalikan database saat operasi backup selesai. TIP : Backup



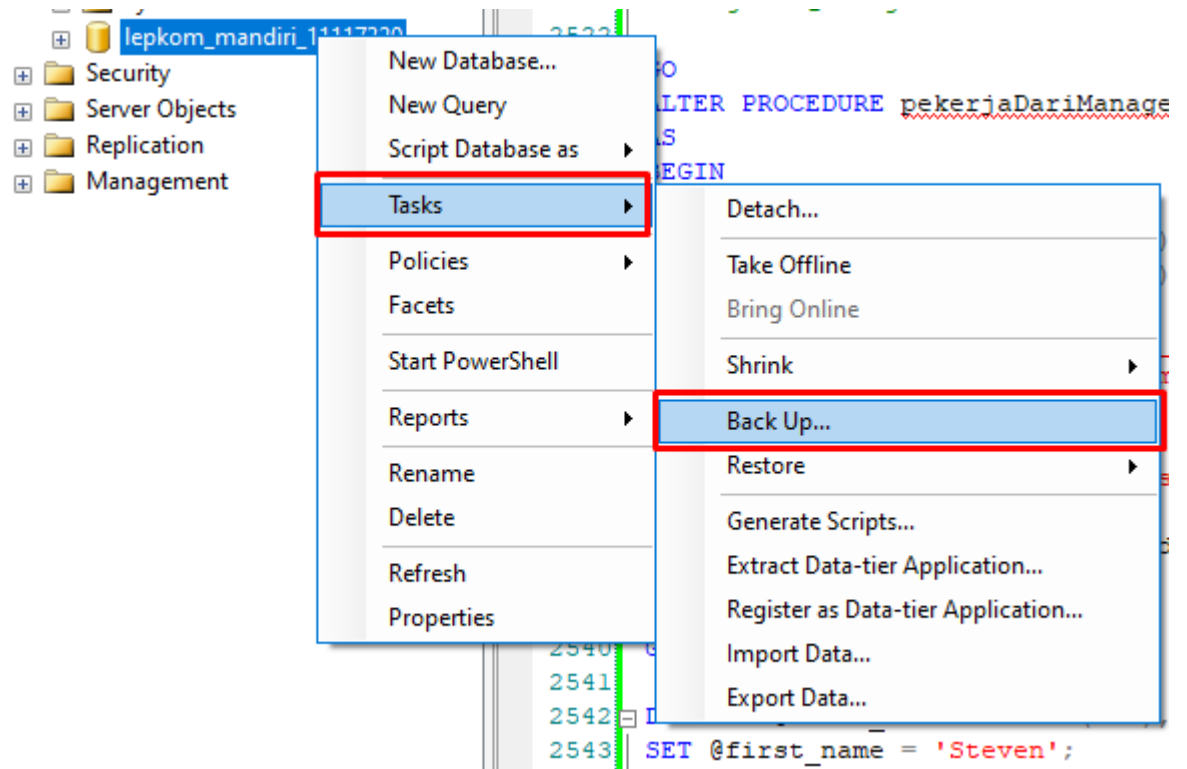
diferensial dapat digunakan hanya dalam hubungannya dengan backup penuh, dan backup diferensial tidak dapat dilakukan pada database master. Jangan bingung antara backup diferensial dengan incremental backup. Backup diferensial merekam semua perubahan sejak full backup terakhir (yang berarti jumlah data yang dibackup tumbuh dari waktu ke waktu). Incremental backup mencatat perubahan sejak backup penuh atau incremental terbaru (yang berarti ukuran data yang dibackup secara bertahap biasanya jauh lebih kecil daripada full backup).

3. Transaction Log Backup. Backup Log transaksi adalah pencatatan serial dari semua modifikasi basis data dan digunakan selama operasi pemulihan untuk commit transaksi yang telah selesai dan untuk roll back transaksi yang belum selesai. Ketika membuat backup log transaksi, backup menyimpan perubahan yang telah terjadi sejak backup log transaksi terakhir dan kemudian memotong log, yang menghapus transaksi yang telah dilakukan atau dibatalkan. Tidak seperti backup penuh dan diferensial, backup log transaksi mencatat keadaan log transaksi pada saat operasi backup dimulai (bukan saat berakhir)
4. File dan filegroup backup. Backup ini memungkinkan membuat backup file database dan filegroup daripada keseluruhan database. Jenis backup ini berguna jika berurusan dengan database besar dan ingin menghemat waktu dengan membackup file individual daripada keseluruhan database. Banyak faktor yang mempengaruhi backup file dan filegroup. Bila menggunakan file dan filegroup backup, harus dicadangkan log transaksi juga, karena ketergantungan ini, maka tidak dapat menggunakan teknik backup ini jika Truncate Log On Checkpoint diaktifkan. Selanjutnya, jika objek dalam database mencakup beberapa file atau filegroup, maka harus membackup semua file dan filegroup yang terkait secara bersamaan.

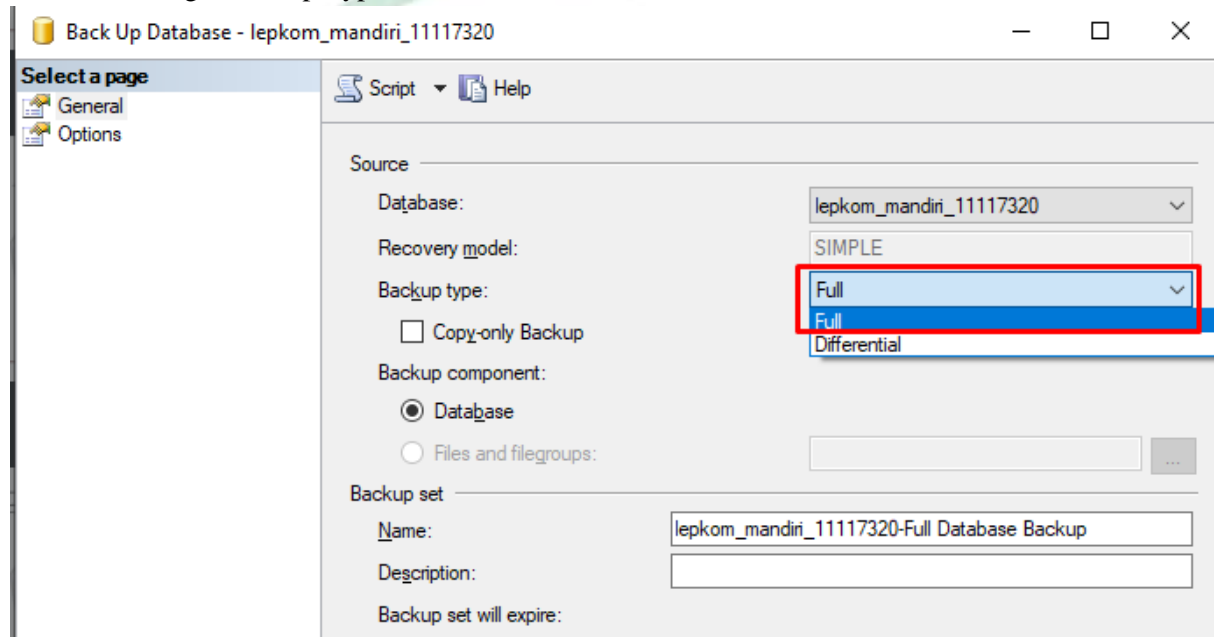
## **2. Langkah-Langkah**

**Lakukan Backup terhadap Database LePkom, untuk semua jenis Backup. Capture dan beri penjelasan langkah-langkah backup tersebut pada File Laporan.**

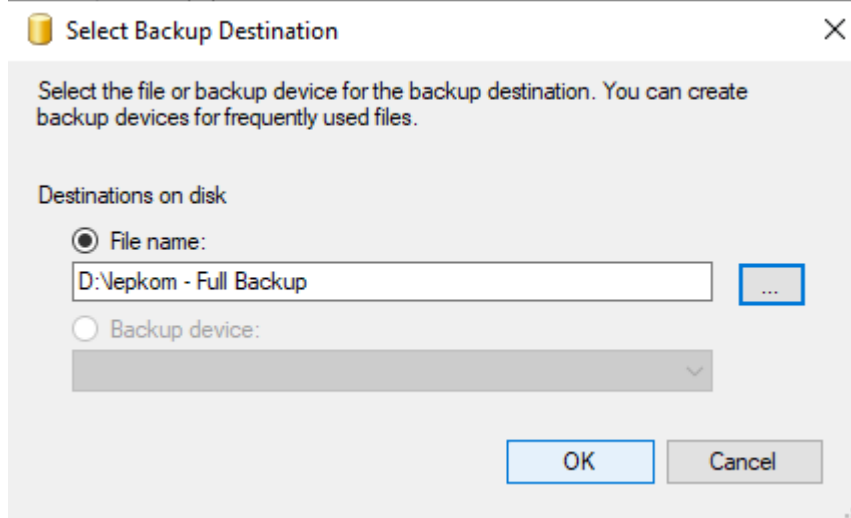
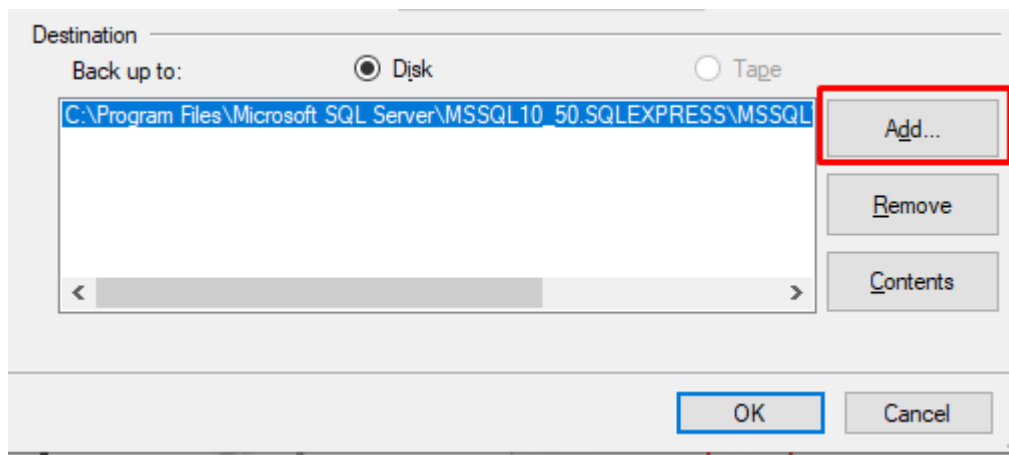
1. Full database Backup Klik Kanan pada Database Task lalu Backup



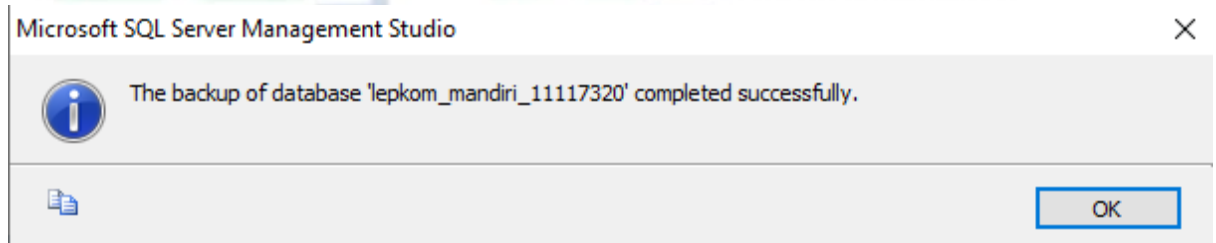
2. Pilih Full Sebagai Backup Type



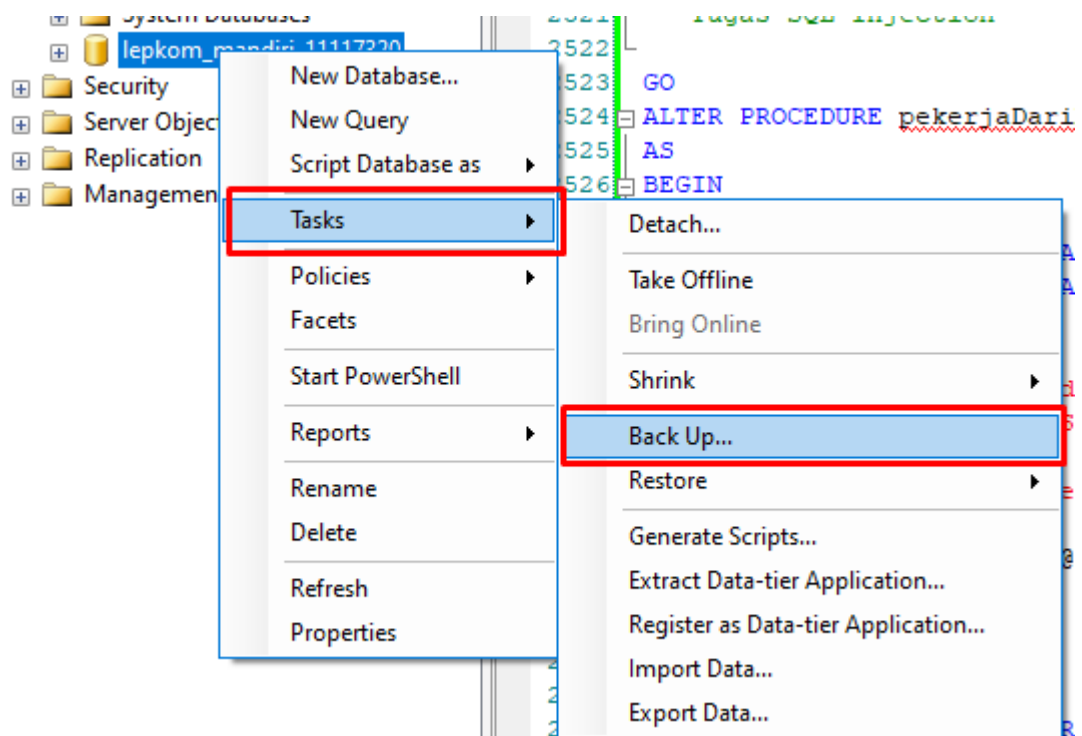
3. Pada Destination Pilih Disk dan tambahkan Folder Dimana ingin meletakkan Backup



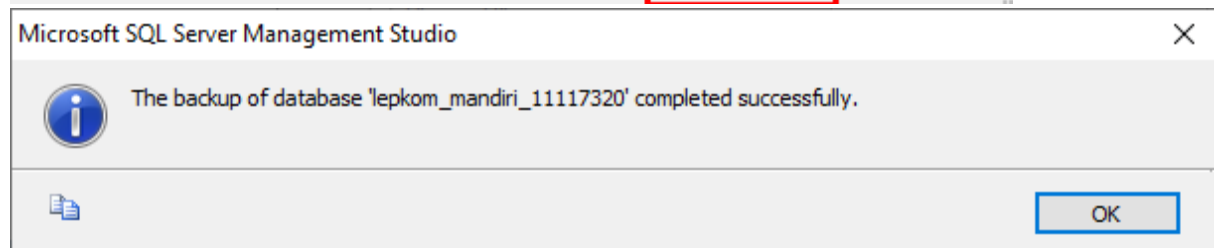
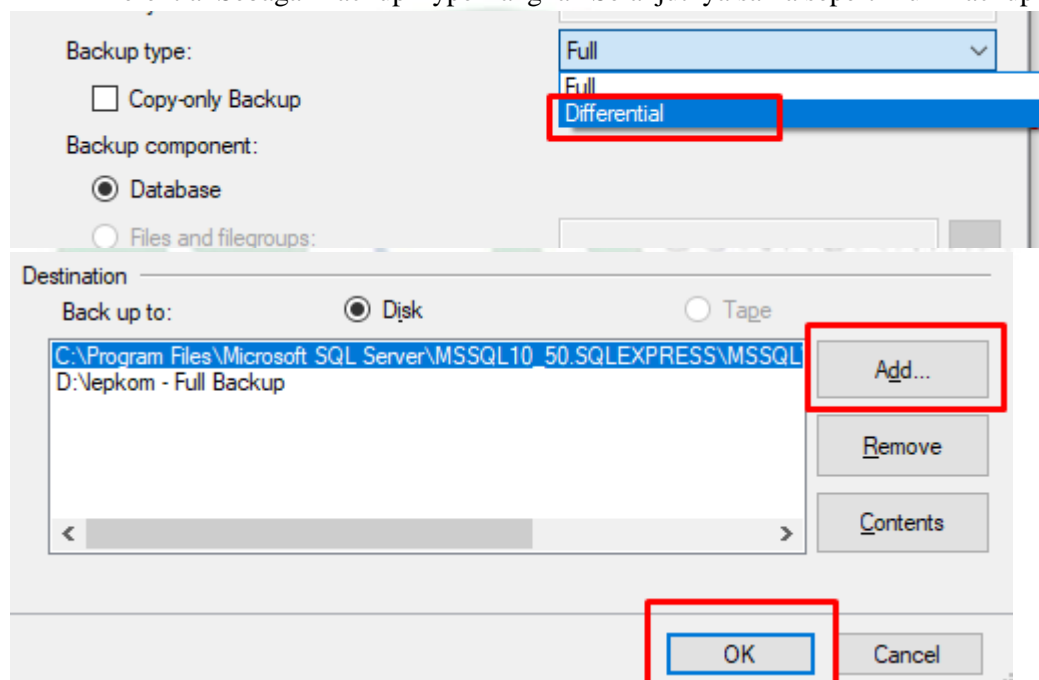
4. Click Ok dan Finish



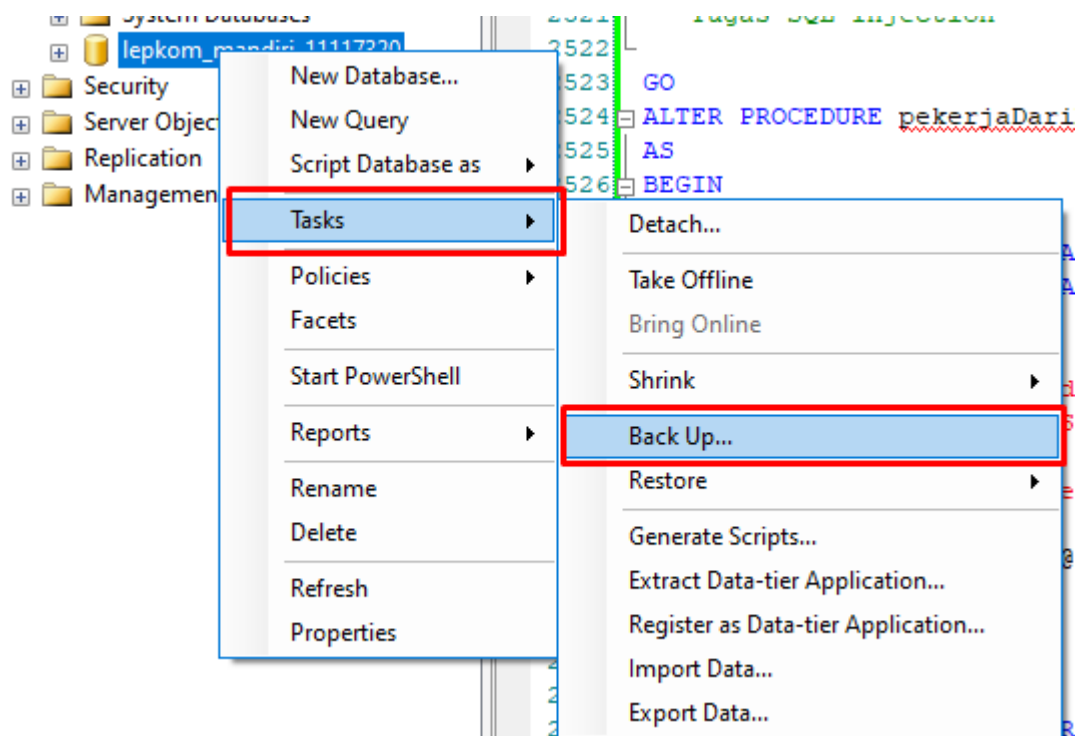
5. Differential Backup (Klik Kanan Task dan Backup)



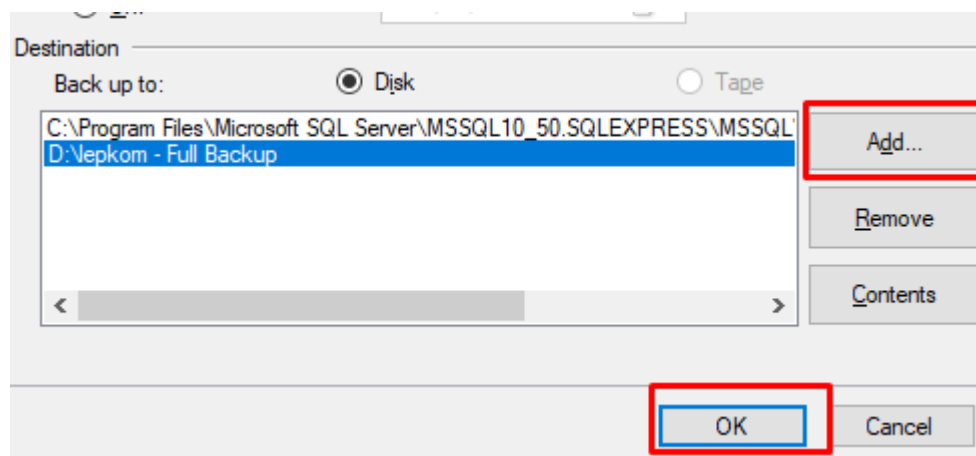
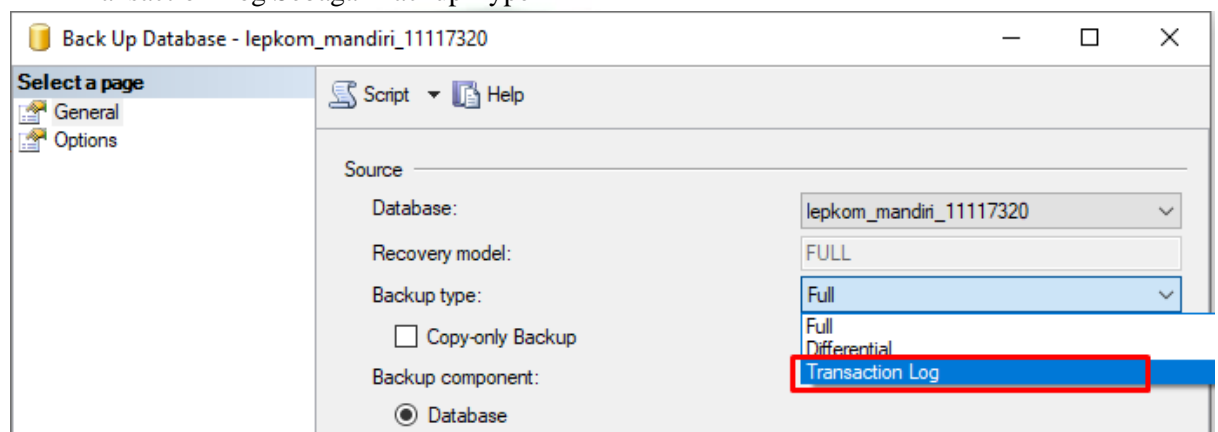
6. Pilih Differential Sebagai Backup Type Langkah Selanjutnya sama seperti Full Backup

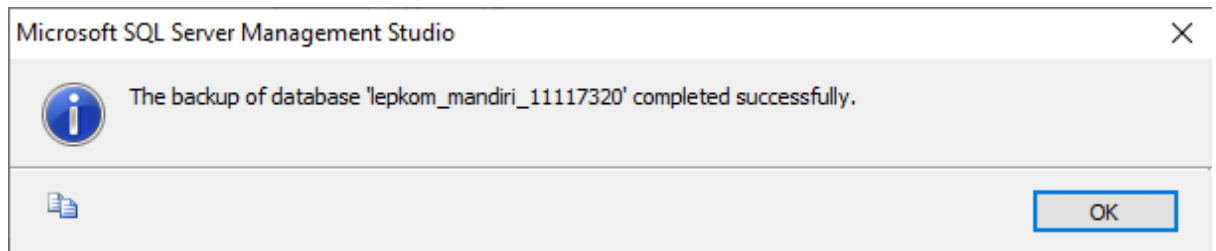


7. Transaction Log Backup (Klik Kanan Database Tasks Backup)

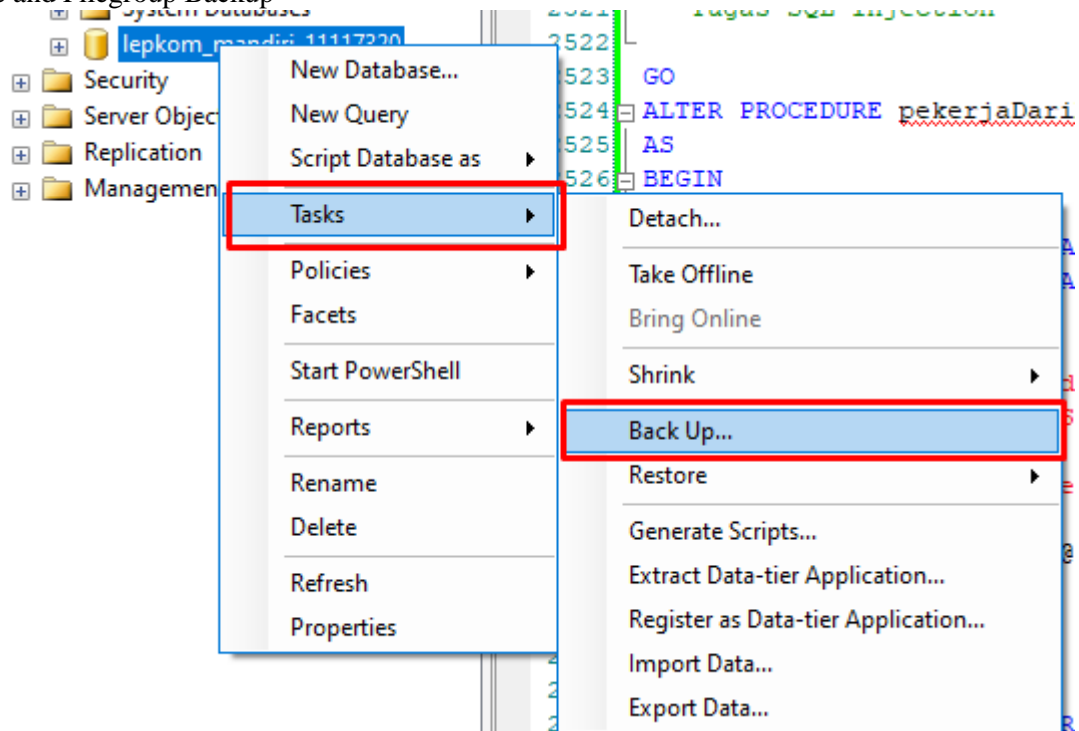


8. Pilih Transaction Log Sebagai Backup Type

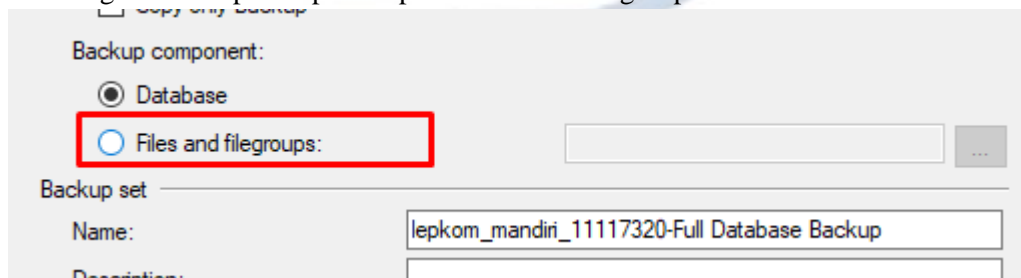




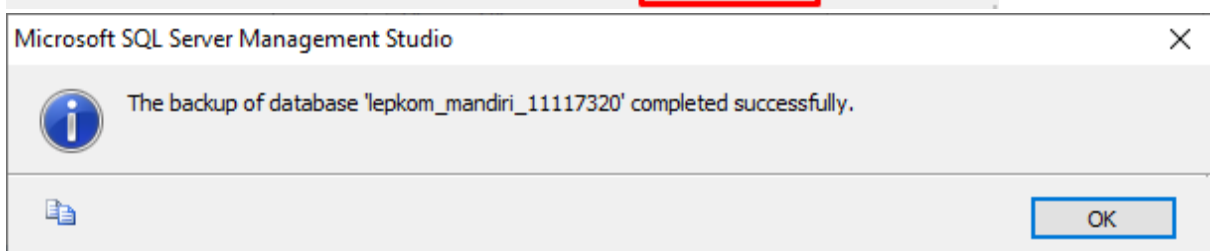
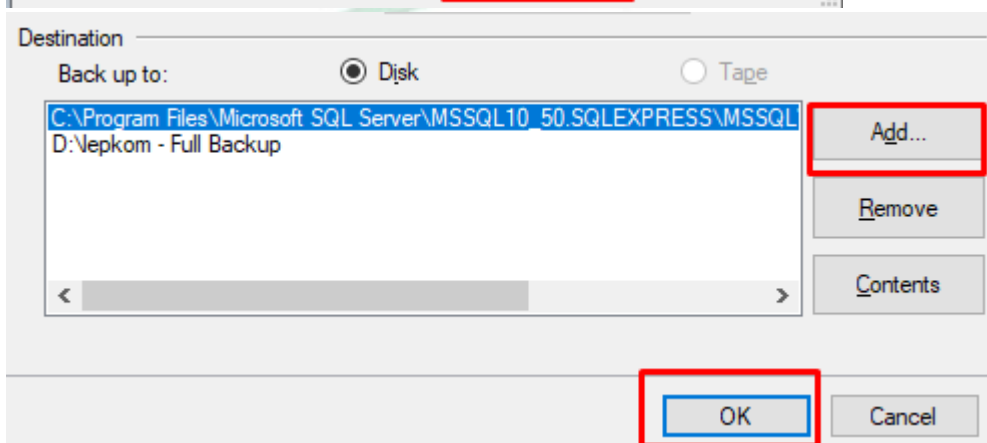
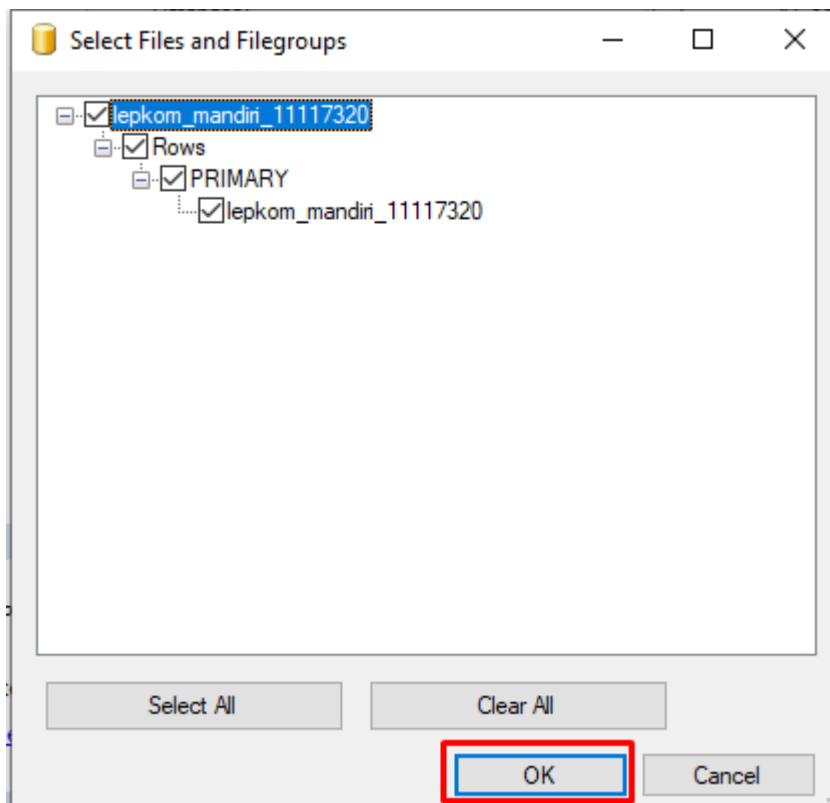
9. File and Filegroup Backup



10. Pada bagian Backup Component pilih Files and Filegroups



11. Checklist sampai ke Table dibawah click Ok



# **PENUTUP**

## **1. Kesimpulan**

Pada praktikum ini telah mempelajari mengenai Backup Tipe tipe Backup dan bagaimana cara melakukan Backup

## **2. SaranPenulisan**





# **LAPORAN AKHIR PROYEK**



PERFORMING DATABASE RECOVERY

Disusun Oleh:

Nama : 11117320

NPM : Chantika Amanda

Kelas :

**LEMBAGA PENGEMBANGAN KOMPUTERISASI  
UNIVERSITAS GUNADARMA**

**2022**

# DAFTAR ISI

DESKRIPSI LAPORAN .....	3
TAHAPAN Pengerjaan.....	4
PENUTUP .....	7



# **DESKRIPSI LAPORAN**

## **PERFORMING DATABASE RECOVERY**

Pada pertemuan 7 laporan berisikan mengenai cara melakukan Recovery dan jenis jenis Recovery



# TAHAPAN Pengerjaan

## 1. Ringkasan Materi

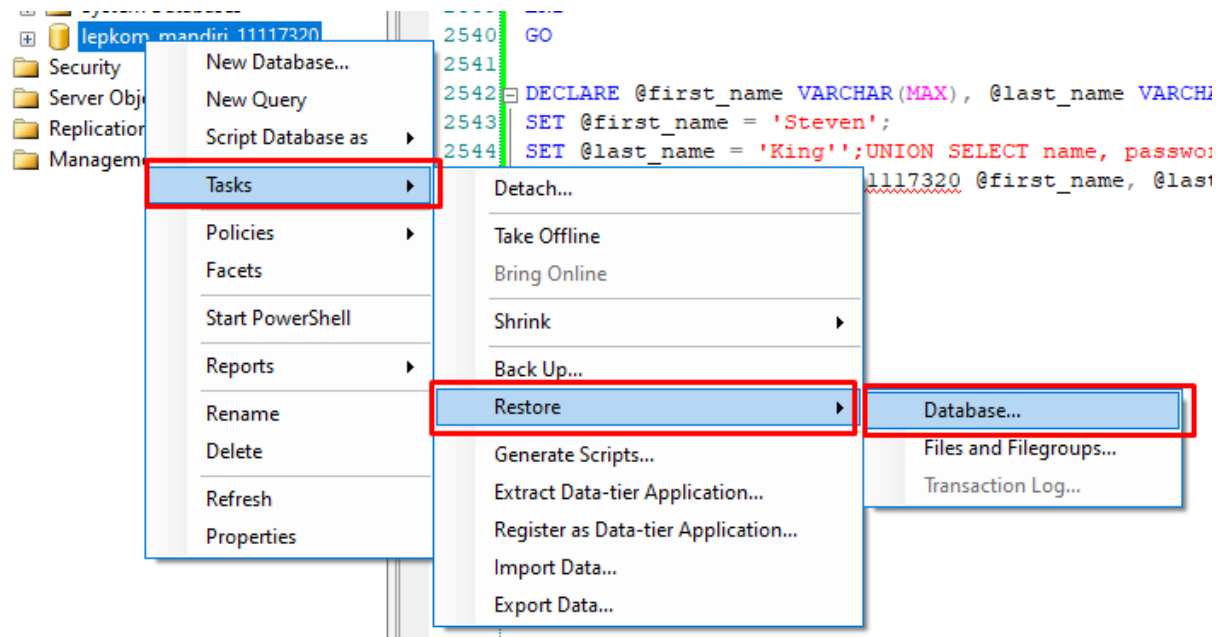
SQL Server 2008 menggunakan model pemulihan untuk membantu merencanakan backup. Jenis database yang dibackup dan jenis backup yang dilakukan mendorong pilihan untuk model pemulihan. Tiga model pemulihan tersedia:

1. Simple. Model pemulihan sederhana dirancang untuk database yang perlu dipulihkan sampai pada titik backup terakhir. Strategi backup dengan model ini harus terdiri dari backup penuh dan diferensial. Backup log transaksi tidak dapat dilakukan saat model pemulihan sederhana diaktifkan. SQL Server 2008 mengaktifkan pilihan Truncate Log On Checkpoint, yang menghapus entri tidak aktif dalam log transaksi di Checkpoint. Model ini membersihkan log transaksi, maka sangat ideal untuk kebanyakan database sistem.
2. Full. Model pemulihan penuh dirancang untuk database yang perlu dipulihkan sampai titik kegagalan atau pada titik waktu tertentu. Bila menggunakan model ini, semua operasi dicatat, termasuk operasi massal dan pemuatan massal data. Strategi backup dengan model ini harus mencakup backup penuh, diferensial, dan transaksi atau backup log transaksi penuh dan hanya log
3. Bulk-logged. Model pemulihan massal-log (bulk-logged) mengurangi penggunaan ruang log namun tetap mempertahankan sebagian besar fleksibilitas model pemulihan penuh. Dengan model ini, operasi massal dan muatan massal minimal login dan tidak dapat dikendalikan secara per operasi. Anda perlu mengulang secara manual operasi massal dan beban massal jika database gagal sebelum Anda melakukan backup penuh atau diferensial. Strategi cadangan dengan model ini harus mencakup backup penuh, diferensial, dan transaksi atau backup log transaksi penuh dan hanya log

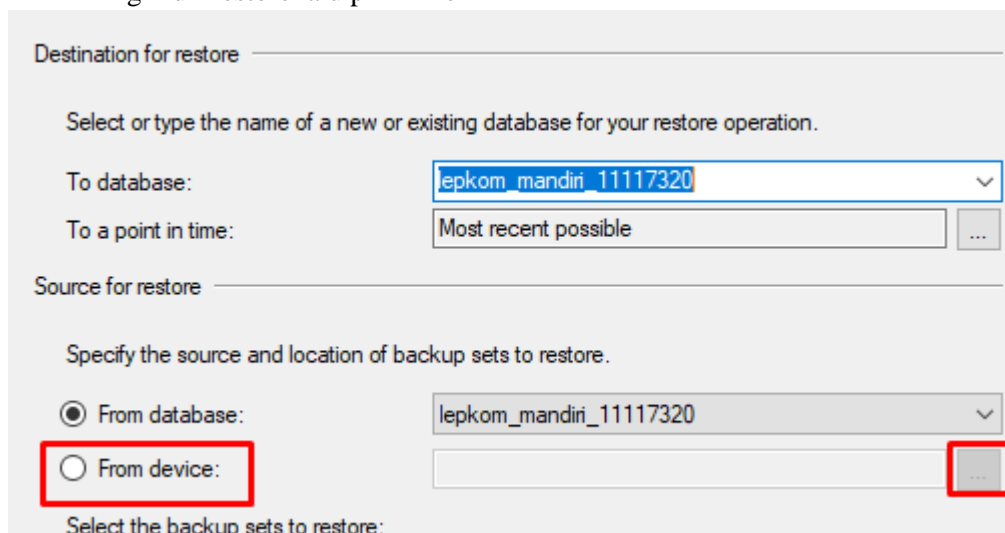
## 2. Langkah-Langkah

**Lakukan Recovery terhadap Database LePkom. Capture dan beri penjelasan langkahlangkah recovery tersebut pada File Laporan.**

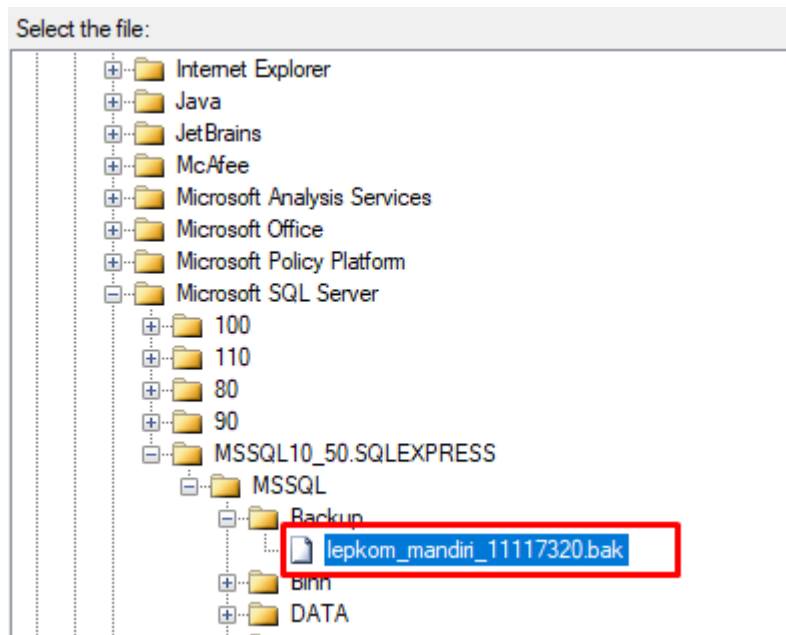
1. Klik kanan pada Database task Restore



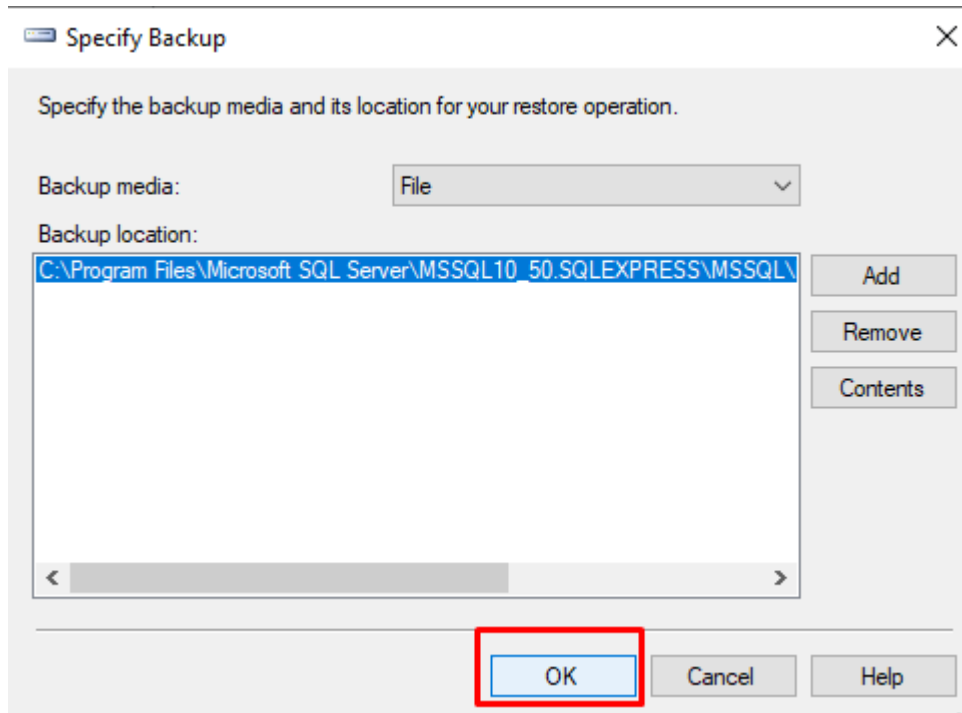
2. Pilih From Device pada Source pastikan Destination sudah betul pada database yang ingin di Restore lalu pilih File



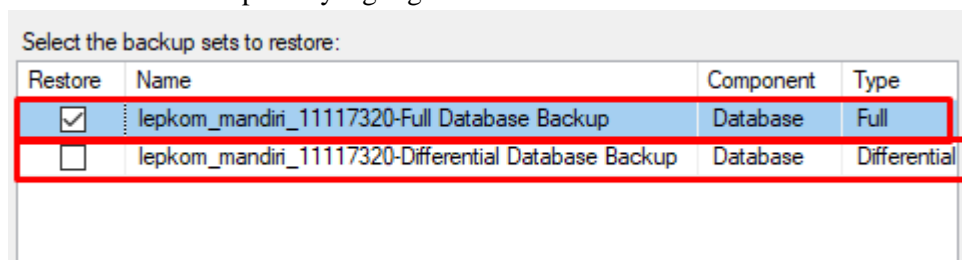
3. Pilih file Bak



4. Click OK



5. Pilih Backup Sets yang ingin di Restore



# **PENUTUP**

## **1. Kesimpulan**

Pada Praktikum ini telah mempelajari mengenai Recovery, Tipe tipe Recovery dan juga bagaimana cara melakukan Recovery pada database

## **2. SaranPenulisan**



# **LAPORAN AKHIR PROYEK**



TRANSFERING DATA

Disusun Oleh:

Nama : 11117320

NPM : Chantika Amanda

Kelas :

**LEMBAGA PENGEMBANGAN KOMPUTERISASI  
UNIVERSITAS GUNADARMA**

**2022**



# DAFTAR ISI

DESKRIPSI LAPORAN .....	3
TAHAPAN Pengerjaan.....	4
PENUTUP .....	12



# **DESKRIPSI LAPORAN**

## **TRANSFERING DATA**

Pada pertemuan 8 laporan berisikan cara melakukan ekspor dan juga cara melakukan Import data



# TAHAPAN Pengerjaan

## 1. Ringkasan Materi

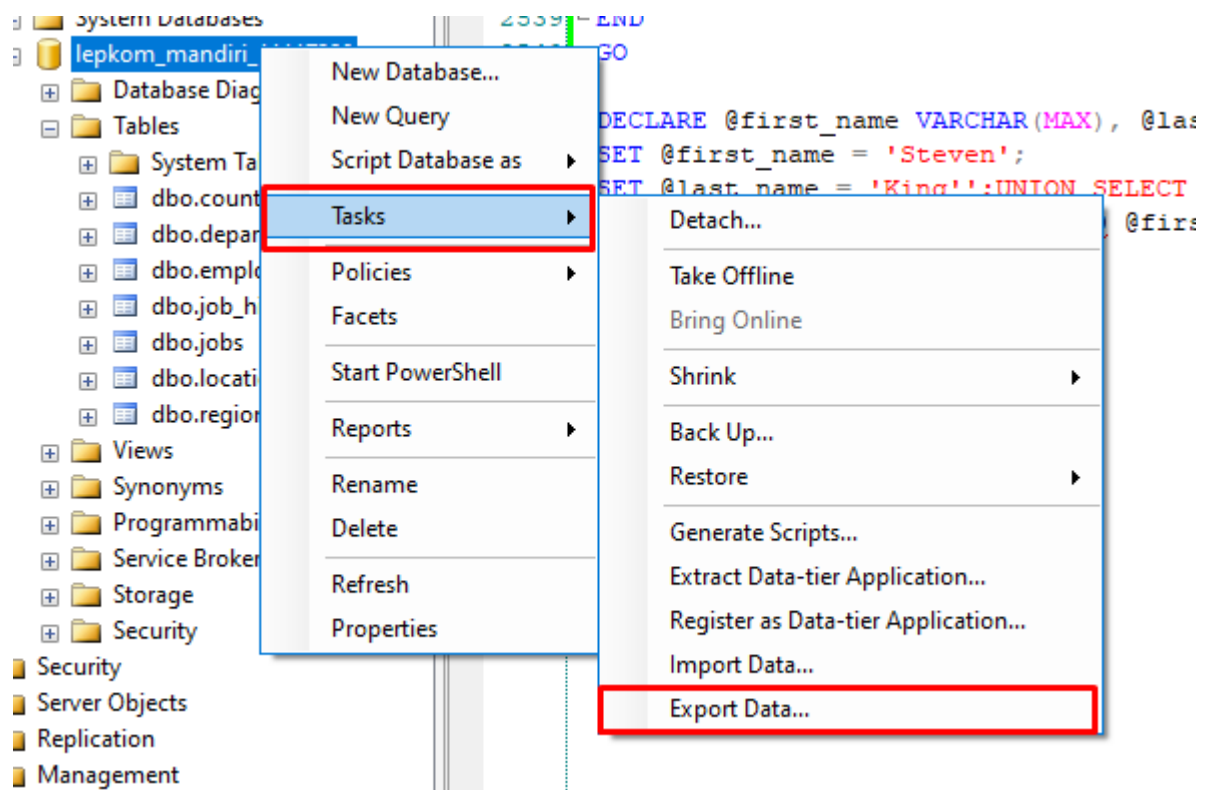
SQL Server Integration Services (SSIS) adalah alat yang digunakan untuk melakukan operasi ETL; yaitu mengekstrak, mengubah dan memuat data. Pemrosesan ETL biasa dilakukan dalam aplikasi data warehousing (DW), namun SSIS sama sekali tidak terbatas hanya pada DW; misalnya ketika membuat Rencana Pemeliharaan menggunakan SQL Server Management Studio (SSMS), sebuah paket SSIS dibuat. Pada tingkat yang lebih tinggi, SSIS menyediakan kemampuan untuk:

1. Mengambil data dari hampir semua sumber.
2. Melakukan berbagai transformasi pada data; misalnya mengkonversi dari satu jenis ke yang lain, dikonversi ke huruf besar atau huruf kecil, melakukan perhitungan, dan lain-lain.
3. Memuat (load) data ke hampir semua sumber.
4. Mendefinisikan alur kerja (workflow)

## 2. Langkah-Langkah

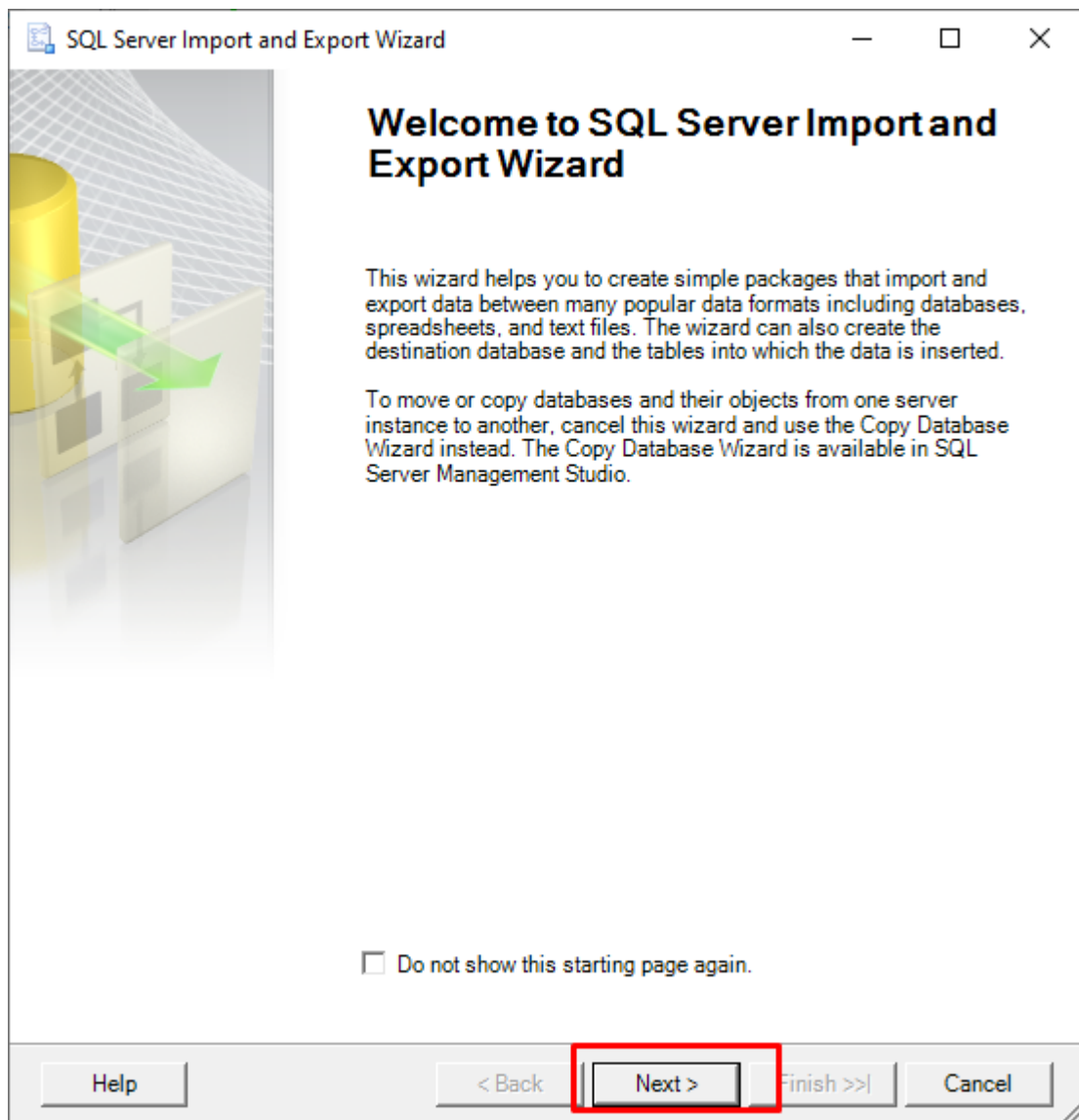
**Lakukan Ekspor terhadap salah satu tabel (pilih). Tempatkan data yang diambil menjadi flat file, dan Microsoft Excel. Capture dan beri penjelasan langkah- langkah ekspor tersebut pada File Laporan.**

1. Txt File (Click kanan pada database task lalu Export Data)

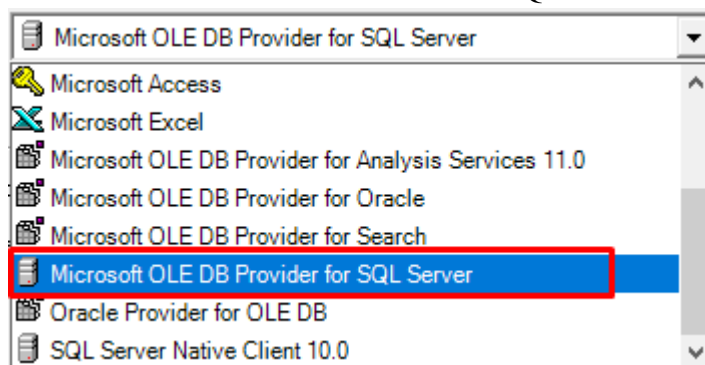


2. Click Next

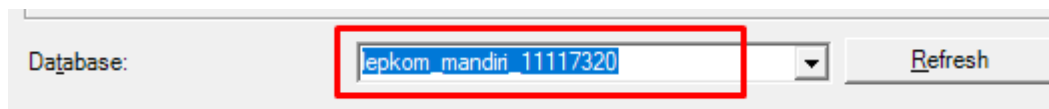




3. Pilih Microsoft OLE DB Provider for SQL Server untuk Source



4. Pastikan Database sudah Benar



5. Pilih Flat File Destination untuk Destination dan Browse untuk masukan File Path tempat Export

Destination: SQL Server Native Client 10.0

Server name:

Authentication:

☒ Use Windows Authentication

☐ Use SQL Server Authentication

User name:

Password:

Flat File Destination

Microsoft Access

Microsoft Excel

Microsoft OLE DB Provider for Analysis Services 11.0

File name:

Browse...

6. Click Export

File name: ExportTXT

Text files (\*.txt)

Open

Cancel

7. Disini bisa pilih yang atas jika mau Export Table atau bawah jika ingin melakukan Query (lebih dari 1 Table)

SQL Server Import and Export Wizard

**Specify Table Copy or Query**

Specify whether to copy one or more tables and views or to copy the results of a query from the data source.

☒ **Copy data from one or more tables or views**

Use this option to copy all the data from the existing tables or views in the source database.

☐ **Write a query to specify the data to transfer**

Use this option to write an SQL query to manipulate or to restrict the source data for the copy operation.

Help

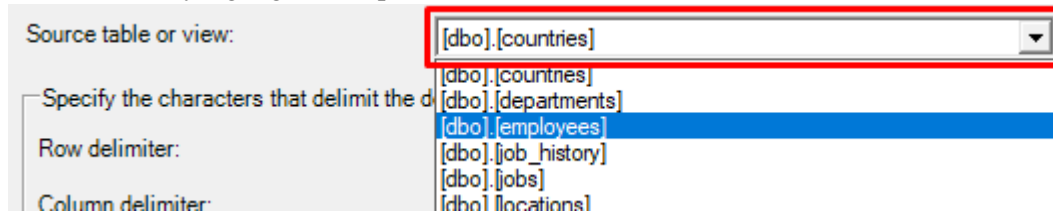
< Back

Next >

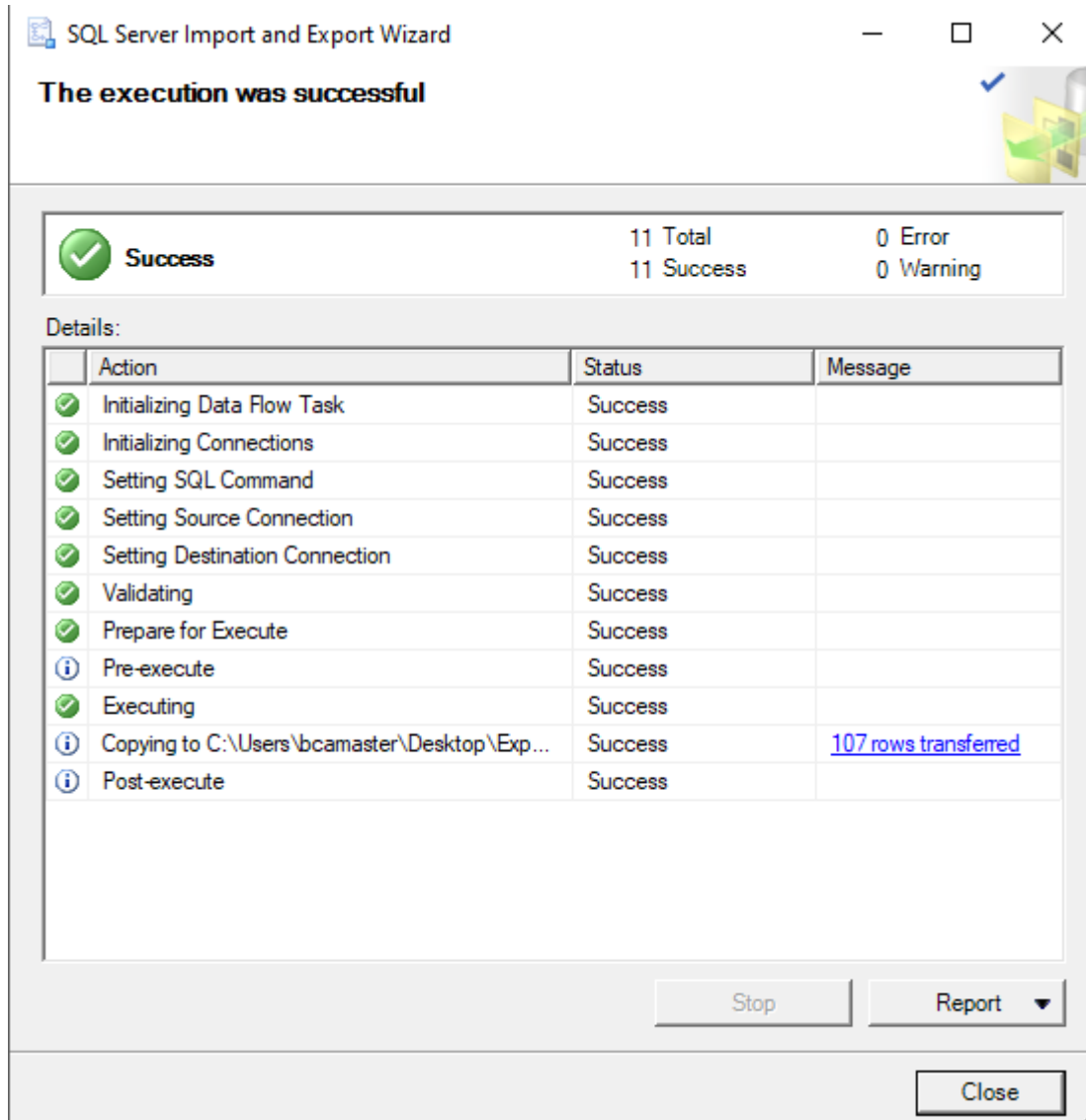
Finish >>

Cancel

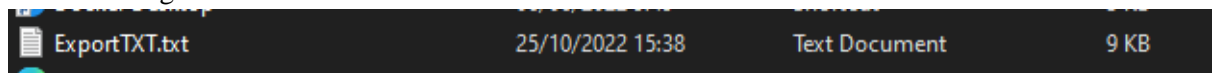
8. Pilih table yang ingin di Export



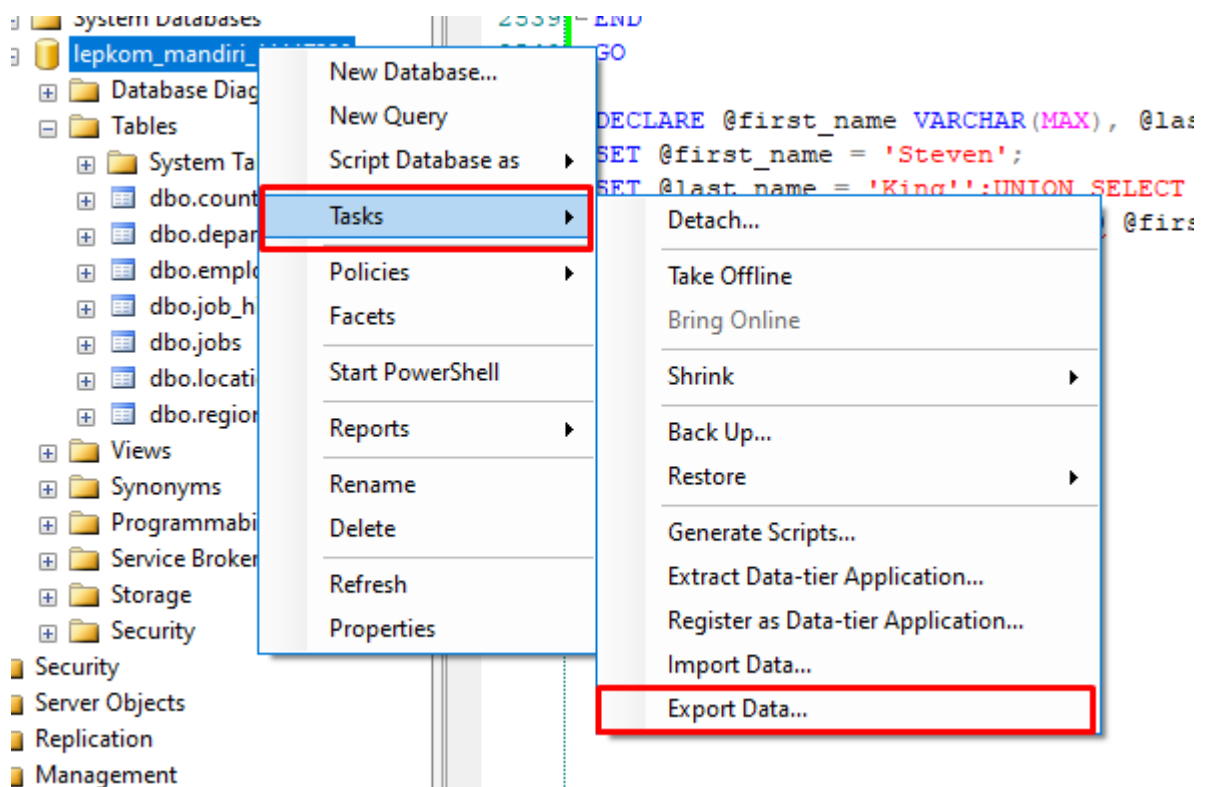
9. Click OK Finish



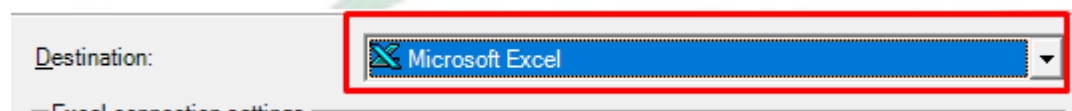
10. Data Tergenerate Sesuai Path



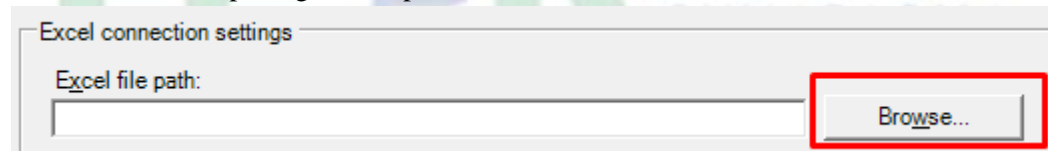
11. Excel



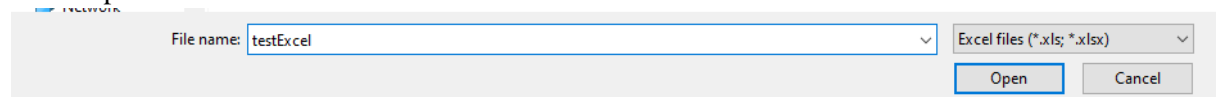
12. Pilih Microsoft Excel untuk Destination



13. Masukkan Path tempat ingin di Export



14. Click Open



15. Centang table yang ingin di Export



**Select Source Tables and Views**

Choose one or more tables and views to copy.



Tables and views:

<input checked="" type="checkbox"/> Source	Destination
<input type="checkbox"/> [dbo].[countries]	
<input type="checkbox"/> [dbo].[departments]	
<input checked="" type="checkbox"/> [dbo].[employees]	'employees'
<input type="checkbox"/> [dbo].[job_history]	
<input type="checkbox"/> [dbo].[jobs]	
<input type="checkbox"/> [dbo].[locations]	
<input type="checkbox"/> [dbo].[regions]	

Edit Mappings... Preview...

Help < Back **Next >** Finish >> Cancel

16. Click Next dan Finish

SQL Server Import and Export Wizard

### Review Data Type Mapping

Select a table to review how its data types map to those in the destination. Also, select how the wizard handles conversion issues.

Table:

Source	Destination
[dbo].[employees]	'employees'

Data type mapping:

	Source Column	Source Type	Destination Co...	Destination Ty...	Convert	On Error
✓	employee_id	numeric	employee_id	Decimal		
⚠	first_name	varchar	first_name	LongText	✓	Use Global
⚠	last_name	varchar	last_name	LongText	✓	Use Global
⚠	email	varchar	email	LongText	✓	Use Global
⚠	phone_NUM...	varchar	phone_NUM...	LongText	✓	Use Global
⚠	hire_date	date	hire_date	varchar	✓	Use Global
⚠	job_id	varchar	job_id	LongText	✓	Use Global

Potential lost conversion from varchar to LongText

To view conversion details, double-click the row that contains the column source type to be converted.

On Error (global) Fail

On Truncation (global) Fail

Help < Back **Next >** Finish >> Cancel

17. Excel akan Tergenerate

testExcel.xls	25/10/2022 15:43	Microsoft Excel 97...	33 KB
---------------	------------------	-----------------------	-------

# **PENUTUP**

## **1. Kesimpulan**

Pada Praktikum ini telah mempelajari bagaimana cara melakukan Export Data pada Database dengan Export Wizard pada SSMS dan menghasilkan File txt dan juga Excel pada Latihan

## **2. SaranPenulisan**

